**Universidad Autónoma Gabriel René Moreno**

Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones

**SISTEMA DE INFORMACIÓN PARA GESTIONAR LAS VENTAS, CURSOS Y ACCESO A LOS DATOS DEL INSTITUTO**

**“INEGAS”**

**GRUPO # ….**

DOCENTE

* Ing. Garzón Cuellar Angélica

INTEGRANTES

* Choque Coca Liz Dara……………………………………………………..217011810
* Torrez Aramayo Jorge Rodrigo……………………………………………217051367

ASIGNATURA

* Sistemas de Información I (INF-342)

FECHA

* 06/05/2019

Santa Cruz – Bolivia

Indice de contenido

1. Perfil…………………………..………………………,,,,,,,,,,,,,,,,,….,,,,,,……………………4
   1. Introducción………………………………………………………………………..4
   2. Objetivos……………………………..……….……………………………………5
      1. Objetivos generales……………………………………………………………...5
      2. Objetivos específicos…………………………………………………..………...5
   3. Antecedentes y estructura organizacional………………………………………6
   4. Justificación………………………………..……………………………………….7
   5. Descripción del problema……………………………………………..…………..8
   6. Formulación del problema……………………………………………………….11
   7. Alcance…….…………………………..………………………………………….12
   8. Entrevistas………………………………………………………………………...14
2. Elementos del sistema basado en computadoras………………………………………....20
   1. Hardware…………………………………………………………………………..20
      1. Servidor…………………………………………………………………...20
      2. Cliente…………………………………………………………………….20
      3. Medios de comunicación………………………………………………..20
      4. Otros dispositivos………………………………………………………...21
   2. Software…………………………………………………………………………...21
      1. Servidor…………………………………………………………………...21
      2. Cliente…………………………………………………………………….21
      3. Otro software adicional…………………………………………………..21
   3. Datos……………………………………………………………………………….21
   4. Procesos………………………………………………………………………….21
   5. Gente/Usuario……………………………………………………………………22
   6. Documento……………………………………………………………………….22
3. Tecnología para el desarrollo del software…………………………………………………23
   1. Estrategia para el desarrollo del software……………………………………..23
   2. Metodología para el desarrollo del software…………………………………..23
      1. Características del PUDS………………………………………………23
      2. Características del UML………………………………………………..25
   3. Herramientas de desarrollo……………………………………………………..26
      1. Software………………………………………………………………….26
      2. Hardware………………………………………………………………...26
4. Posibles costos……………………………………………………………………………….27
5. Posibles beneficios…………………………………………………………………………...27
   1. Tiempo……………………………………………………………………………27
   2. Esfuerzo……………………………………………………………………….....28
   3. Costos…………………………………………………………………………….28
6. Posibles clientes……………………………………………………………………………...28
7. Modelo de dominio……………………………………………………………………………29
   1. Diseño lógico……………………………………………………………………..29
      1. Diagrama de clases…………………………………………………….29
      2. Mapeo……………………………………………………………………30
      3. Normalización…………………………………………………………...32
   2. Diseño físico……………………………………………………………………...33
      1. Tabla de volumen……………………………………………………….33
      2. Script……………………………………………………………………..40
      3. Diagrama relacional…………………………………………………….44
      4. Actualización de tuplas…………………………………………………45
      5. Consultas………………………………………………………………..48
      6. Procedimientos almacenados…………………………………………54
      7. Disparadores…………………………………………………………….57
8. Modelo Ishikawa………………………………………………………………………………61
   1. Identificar el problema…………………………………………………………..61
      1. Lista de problemas...……………………………………………………61
      2. Depurar problemas……………………………………………………..62
      3. Propietarios del problema….…………………………………………..63
      4. Análisis del problema…………………………………………………...64
      5. Estimación y cuantificación de problemas……………………………65
      6. Alternativas de cambios………………………………………………..66
      7. Conclusión de los problemas…………………………………………..66
      8. Modelo Ishikawa………………………………………………………...67
9. Modelo de Negocio…………………………………………………………………………...68
   1. Diagramas de Actividad…………………………………………………………68
10. Captura de Requisitos………………………………………………………………………..71
    1. Encontrar Actores y Casos de Uso…………………………………………….71
       1. Lista de Actores…………………………………………………………71
       2. Lista de Casos de Uso…………………………………………………71
    2. Priorizar Casos de Uso…………………………………………………………73
11. Anexos…………………………………………………………………………………………74

**PERFIL**

* 1. **Introducción**

El instituto para la excelencia en los negocios del gas, energía e hidrocarburos está enfocado en prestar servicios de capacitación a personas y profesionales en los diferentes cursos ofrecidos. INEGAS inició actividades usando un sistema manual de registro de datos de sus operaciones, el cual no es muy eficiente, pero a su parecer cubría las necesidades básicas en sus inicios.

Tiempo después el instituto fue creciendo y con él el número de clientes fue aumentando, lo que implicó en un aumento en las inscripciones que a su vez requerían ser gestionados de una forma más eficiente, cosa que no se pudo dar hasta el momento y cada vez es más urgente la obtención de un sistema que pudiera cumplir con sus expectativas dado su constante crecimiento.

En el presente proyecto llevaremos a cabo el desarrollo de una de las soluciones, enfocada en los procesos de venta, gestión de cursos, gestión de clientes y digitalización de la información de una empresa orientada a la educación de profesionales.

El desarrollo de lo previamente planteado se intentará lograr mediante el uso de diversos métodos de recolección y análisis de datos, de los cuales se esperan resultados satisfactorios que cumplan con las expectativas tanto del grupo del trabajo y en especial las expectativas del cliente. A su vez con todos los datos obtenidos se crearán a partir de ellos un sistema de información que estará orientado al instituto INEGAS.

La finalidad de este proyecto es la mejora en el proceso de trabajo del instituto, así como también un mejor manejo para sus trabajadores, tanto en el área laboral como en el área académica. Al tener una mejor comunicación entre los docentes y los trabajadores, también será accesible una mejora de docentes con alumnos, en función a los cursos que tomen. Lo primordial sería la agilización en los procesos de registro en alumnos nuevos de INEGAS y en los registros del personal administrativo y académico.

* 1. **Objetivos**
     1. **Objetivos generales**

Desarrollar un sistema de información para la gestión de ventas, cursos y acceso a los datos del Instituto para la excelencia en los negocios del gas, energía e hidrocarburos (INEGAS)

* + 1. **Objetivos específicos**
* Recolectar información acerca de INEGAS, utilizando

técnicas como la redacción de un cuestionario para desarrollar entrevistas.

* Analizar la información que se ha recolectado en las entrevistas para

determinar los elementos y funciones de nuestro sistema.

* Diseñar un sistema apropiado para solucionar los problemas existentes en el

área de ventas, cursos, alumnos y personal, el cual se realizará en base al

Lenguaje Unificado de Modelado (UML).

* Implementar la base de datos, la misma que se desarrollara en distintos

gestores de base de datos.

* Implementar la interfaz del software acorde a los requerimientos del usuario.
  1. **Antecedentes y estructura organizacional**

El Instituto para la Excelencia en los Negocios de Gas, Energía e Hidrocarburos (INEGAS) es el resultado de la visión de una Institución Educativa líder: la Universidad Autónoma Gabriel René Moreno (UAGRM), representada en ese momento por Julio Salek Mery, en su calidad de rector, y una empresa de prestigio mundial: Petrobras Bolivia S.A. representada por José Fernando de Freitas, en su condición de Presidente.

Es así que, el 23 de marzo del 2005 se firmó el Acuerdo Marco de Cooperación y el Convenio Interinstitucional, que tenía por objetivo viabilizar la implementación de un Instituto especializado en los negocios de hidrocarburos y energías.

Para formalizar el acuerdo, la UAGRM, emite la RR Nº 135/2005, del 13 de Junio del 2005, que establece ceder una superficie de 10.000 m2 en los predios de la Universidad para que Petrobras pueda encargarse de la construcción y equipamiento del Instituto.

Un aspecto muy importante en el surgimiento de INEGAS, ha sido la definición en el Convenio, donde se establece la conformación del directorio, compuesto por el rector de la UAGRM como Presidente, el Presidente de Petrobras como VicePresidente de la Institución académica, un docente con vasta experiencia y conocimiento en el sector de hidrocarburos de la UAGRM como director, el Gerente de Recursos Humanos de Petrobras como director y de manera independiente el Gerente General de INEGAS.

Hoy, INEGAS es la mayor red de capacitación de post-grado, sus más de 110 unidades en 6 países lo demuestran.

* 1. **Justificación**

La implementación de un sistema de información en el Instituto “INEGAS” se vio necesario debido a que agilizaría el proceso de gestión de datos e información de los clientes, cursos, ventas y personal que actualmente son respaldados mediante papeles, que son constantes problemas que ha estado llevando la empresa, permitiendo así un mejor manejo de los registros necesarios.

* 1. **Descripción del problema**

Un Sistema de información es desarrollado para ayudar a una empresa a mejorar, organizarse y sobresalir en diferentes aspectos. Para comenzar a desarrollar un sistema de información debe existir una razón o problemática que ayude a entender mejor el fin al que se quiere llegar.

El instituto se dedica a dar cursos de diferentes tipos a personas con ganas, esfuerzo y voluntad para aprender y mejorar su desarrollo en su campo laboral. El motivo de su fundación no fue la expansión, pero al ver que se les dió la oportunidad no dudaron en llevarla a cabo, consiguiendo así estar en un top de los mejores institutos en su área.

Se requiere sistematizar el área de ventas y así mismo tener un control sobre el plantel de docentes y estudiantes, ya que actualmente los registros realizados de forma manual son obsoletos e ineficientes para el control de una empresa creciente.

De los docentes se desea registrar los tipos de cursos que dan, así como también sus años de experiencia y profesión del docente.

Detallaremos los procesos que realiza la empresa:

* Análisis de los docentes

Se realiza un control de calidad del docente el cual será determinado acorde al tipo de curso que se desea elaborar, en caso de no cumplir a cabalidad con los estándares establecidos por la empresa, estos son descartados o designados para otros propósitos.

Todo este proceso se lleva a cabo en instalaciones de la empresa, controlando de esta manera las características del curso obtenido.

* Realización de los cursos

Una vez analizados los docentes, se procede a la realización de los cursos.

Los procesos de organización son sencillos y muy cuidadosos, llevados a cabo cumpliendo las directrices de un sistema de calidad riguroso.

Se cuida todo el proceso de gestión de cursos de forma global, del principio al final, empezando por la selección de cursos y docentes hasta la calidad del curso terminado.

* ***Personal***

Actualmente el instituto no cuenta con un sistema encargado de algún área, se trata de un instituto que solo está organizado en docentes temporales y administrativos.

Como una proyección a futuro se quiere controlar el registro de personal para el instituto, como ser más administradores, docentes y alumnos, entre otros. Así mismo se quisiera tener un seguimiento de las funciones de todo el personal del instituto.

Con la implementación del sistema se desea poder tener un registro de este personal contratado con información esencial como ser: Nombre completo, ci, fecha de ingreso, edad, etc. Tener un registro de las ventas que se han realizado. También poder tener un registro de los alumnos del instituto, divididos por algún tipo de prioridad como la antigüedad. Todos los empleados, docentes y alumnos deberán tener un código para poder especificar a cada persona.

Para el área de los docentes, la empresa no tiene un sistema para almacenar los datos más importantes de éstos. Un objetivo para el final de este proyecto, es que el instituto pueda administrar mejor su platel docente mediante un buen sistema y adaptable para que lo puedan usar sin mayores complicaciones.

* ***Inventario***

El inventario sería todos los tipos de cursos que ofrece el instituto, cada uno de los diferentes avances y sus respectivos docentes.

Todo este inventario es monitoreado y controlado por gerente general, y otros trabajadores encargados del área.

Este proceso es llevado a cabo de forma analógica (papeles) en el cual se pierde mucho tiempo y energía al no ser un sistema. Se va logar poner el inventario para así lograr tener un control sistemático y facilitar el registro de nuevos cursos o adición de los mismos, tener una vista de qué cursos están disponibles, y así mismo facilitar la búsqueda de cada uno, tanto para el administrador como para los alumnos.

* ***Servicios***

Venta

La realización de los cursos se lleva a cabo en el mismo instituto. Anticipadamente el cliente hace la inscripción en el curso o cursos que desea tomar. El control de las ventas se lo realiza de forma manual en un libro diario que tienen a disponibilidad las propietarias.

Se desea tener un control sistemático de estas ventas, ya que el control manual resulta más moroso e ineficiente para el instituto.

Siendo tantas las variables que afectan en el desenvolvimiento de la empresa, hemos de identificar los eslabones más débiles, para fortalecerlos mediante el software a implementar; en estos citamos:

* Control de Inventario
* Ventas
* Registros de alumnos y personal de la empresa
* Acceso a la información en cualquier momento
* Inconsistencia de datos
* Mejor comunicación alumno – instituto
  1. **Formulación del problema**

Debido a que hasta el momento los trabajadores del Instituto instituto para la Excelencia en los Negocios del gas, energía e hidrocarburos “INEGAS” continúa realizando de forma manual algunos procesos que se pueden automatizar, se vio la necesidad de desarrollar un sistema de información que permitiera gestionar datos e información de los alumnos, personal, cursos y ventas, agilizando todo el proceso y hacerlo más eficiente.

* 1. **Alcance**

El instituto no cuenta con un sistema de información que le permita registrar a los clientes, empleados. Se genera muchas dificultades a la hora de procesar los datos en términos de tiempo y las posibles falencias que este trabajo manual implica, sin mencionar el riesgo de que estos se pierdan. La información acerca de los clientes, como sus datos personales, servicios que adquieren, preferencias, etc. Son datos que el instituto desconoce, genera una mala interacción cliente-instituto, el cual puede causar problemas al instituto. A continuación, detallamos los módulos en los cuales se generan dificultades:

**Gestionar cursos**

Esta gestión se administrarán todos los cursos que ofrece el instituto, cada uno de los diferentes avances y sus respectivos docentes.

Todo este inventario es monitoreado y controlado por el gerente general y trabajadores del área, de los cuales se desea guardar:

* Código, nombre, duración (entre los más importantes)

**Gestionar tipos de cursos y docentes**

Se llevará un registro de los diferentes tipos de cursos que ofrece el instituto y a su vez, estos tipos de cursos también servirán para clasificar a los tipos de docentes que hay en el instituto.

* Identificador, nombre

**Gestionar Alumnos**

* **Registrar alumno:** Cuando un nuevo alumno quiera inscribirse y no esté en el sistema, el sistema le asignará automáticamente un código y se guardarán los siguientes datos:
  + Código, nombre, CI, teléfono, correo electrónico
* **Modificar alumno:** teniendo algún problema en el registro o algún cambio de sus datos se precederá a modificar con una previa verificación.

**Gestionar docentes**

Se tendrá un registró de los docentes que dan los cursos ofrecidos en función a los grupos en los que se encuentra cada curso y su gestión.

* Código, nombre, profesion

**Módulo de Administración de personal**

En este módulo se registrará toda la información concerniente al personal que trabaja dentro de la empresa y sus respectivos cargos.

**Gestionar empleado:**

* **Registrar empleado:** Cuando un nuevo empleado sea contratado y no esté en el sistema, el sistema le asignará automáticamente un código y se guardarán los siguientes datos:
* Código, nombre, años de experiencia
* **Modificar empleado:** teniendo algún problema en el registro o algún cambio de sus datos se precederá a modificar con una previa verificación.

**Registrar Departamentos y cargos:**

Se podrá registrar los cargos y departamentos que están vigentes en el instituto guardándolos con los siguientes datos:

* Identificador, nombre, estado

**Registrar Ventas:**

Las ventas serán guardadas en la base de datos mediante notas de venta, de estas se requiere saber:

- Nro, fecha, monto, persona que realiza la venta, persona a la que se vende, curso que se está vendiendo

**Controlar accesos de trabajadores:**

No todos los trabajadores tendrán un acceso a todos los datos del sistema, cada trabajador tendrá una cuenta de usuario y esta a su vez tendrá privilegios para poder realizar ciertos cambios en el sistema. En cada usuario se tendrá:

-Nombre de usuario, correo, contraseña

* 1. **Entrevistas**

Las entrevistas se realizaron a unos cuantos trabajadores de INEGAS en las fechas 09/04/2019 y 10/04/2019 en los mismos ambientes de la empresa ubicados en la avenida Bush entre segundo y tercer anillo, a continuación, veremos la entrevista que se realizó:

**Entrevista para obtención de requisitos**

**Objetivo:**

**Entrevista # 1**

**Lugar:** Av. Bush entre segundo y tercer anillo

**Duración:** 45 minutos

**Datos de la Empresa**

**Nombre:** INEGAS

(**X**)Privada (**X**)Estatal

**Datos del Entrevistado**

**Nombre:** Jorge Torrez Alvarez

**Cargo:** Coordinador en marketing y publicidad

**Datos de los Entrevistadores**

**Nombres**: Liz Dara Choque Coca

**1. ¿Cómo se llama la empresa?**

La empresa se llama INEGAS que significa Instituto para la excelencia en los negocios del gas, energía e hidrocarburos. Es una unidad desconcentrada de la universidad autónoma Gabriel René Moreno, que genera sus propios recursos.

**2. ¿En dónde guarda los registros de ventas?**

Los registros de venta son guardados en un sistema contable.

**3. ¿Cómo ofertan sus cursos?**

Se ofertan vía Facebook, LinkedIn, Whatsapp, e-mail, y vía llamadas.

**4. ¿Cuenta con personal suficiente la empresa?**

Si, contamos con personal suficiente.

**5. ¿Para que una persona entre a trabajar en la empresa se solicita experiencia?**

Si, preferiblemente es necesaria la experiencia.

**6. Si algún cargo lo requiere, ¿tienen capacitaciones para nuevos trabajadores?**

No, no hay capacitaciones para los nuevos, pero si hay una inducción.

**7. ¿La empresa cuenta con sucursales en la ciudad?**

No, no hay sucursales.

**8. ¿La empresa cuenta con un sitio web?**

Si, el sitio web es: [www.inegas.edu.bo](http://www.inegas.edu.bo)

**9. ¿Qué se necesitaría para mejorar la empresa?**

Un sistema que agilice los procesos de datos y registros de estudiantes.

1. **¿A qué se dedica la empresa?**

Ofrece capacitación en cursos de capacitación continua y a nivel de postgrado.

1. **¿Cómo se maneja el inventario de los cursos?**

Los cursos son elaborados por el área académica, luego promocionada para la comercialización por el área de marketing o comercial.

Realizado el curso, este generará notas y datos de estudiantes, los cuales se almacenan en físico (documento) y en un sistema de datos del área académica.

**12. ¿Cuáles son los cargos que existen en la empresa?**

Los cargos son los siguientes:

Gerente general, jefe administrativo financiero, coordinador académico e investigación, jefe comercial y marketing, contador, auxiliar contable, responsable de compras y adquisiciones, responsable de recursos humanos, encargado de caja, mensajero, asesor legal, asistente legal, encargado de logística, encargado de alquiler de ambientes, secretaria, encargado de seguimiento académico, responsable de capacitación continua, auxiliar académico, responsable de programas de postgrado, encargado de marketing, encargado de ventas, responsable de proyectos institucionales, atención al cliente y auxiliar comercial.

1. **¿Cuál es el horario de atención de la empresa?**

El horario administrativo es de lunes a viernes de 8:00 a 12:00 y de 15:00 a 18:30.

El horario académico de lunes a viernes es de 8:00 a 12:00 y de 15:00 a 22:00, los sábados es de 8:00 a 12:00 y de 15:00 a 20:00, y finalmente los domingos de 8:00 a 13:00.

**14. En el tema de marketing, ¿Que sistemas o medios de comunicación utilizan para difundir al instituto?**

INEGAS difunde la publicidad vía Facebook, LinkedIn, Whatsapp, e-mail, y medios impresos.

**15. ¿Cómo se realizan los registros de los nuevos alumnos?**

Los registros de los nuevos alumnos se los realiza de dos maneras:

*Vía caja de INEGAS*, el prospecto presentará su carnet de identidad el encargado de ventas para iniciar el llenado de formulario de preinscripción. Luego se apersonará por caja con el formulario para realizar el pago por el curso indicado y el registro en el asistente contable.

*Vía banco*, el prospecto deberá realizar el pago en la cuenta de banco de INEGAS. Luego con el comprobante deberá apersonarse por el departamento de marketing para hacer el formulario de preinscripción y con ese documento se apersonará por caja para hacer el registro en el sistema contable.

**Entrevista para obtención de requisitos**

**Objetivo:**

**Entrevista # 2**

**Lugar:** Av. Bush entre segundo y tercer anillo

**Duración:** 2 horas

**Datos de la Empresa**

**Nombre:** INEGAS

(**X**) Privada (**X**) Estatal

**Datos del Entrevistado**

**Nombre:** Vivian Avila

**Cargo:** Responsable de social media

**Datos del Entrevistador**

**Nombres**: Jorge Rodrigo Torrez Aramayo

**1.- ¿Cuánto es el ingreso mensual de la empresa?**

No puedo proveer esa información sobre la empresa.

**2.- ¿Cuándo se creó la empresa?**

Se creó hace 13 años.

**3.- ¿Cuántos empleados tiene la empresa?**

Tiene 25 aproximadamente.

**4.- ¿Esta empresa cuenta con ambientes propios?**

Si, tenemos ambientes propios.

**5.- ¿Cuál fue el año donde la empresa genero mayores ingresos?**

No tengo conocimiento sobre los ingresos por año.

**6.- ¿La empresa cuenta con vehículo?**

Tenemos una moto para mensajería.

**7.- ¿La empresa tiene estacionamiento propio?**

Contamos con un parqueo propio.

**8.- ¿Cuáles son los tipos de alumnos a los que llegan?**

Nuestro público objetivo y consumidor son los profesionales en áreas de tecnología, a los que les interese realizar programas de postgrado.

Por el área de cursos cortos de capacitación continua el público objetivo son profesionales específicos para cursos específicos, igualmente personas con gran deformación técnica o empírica.

**9.- ¿Cuáles son los tipos de cobranza que tiene la empresa?**

Las cobranzas son en efectivo en caja de INEGAS y también vía cuenta de banco.

**10.- ¿Cómo registran sus compras, ventas y servicios?**

Compras ventas y servicios son registrados en un sistema contable.

**11.- ¿Se brinda asesoramiento a los alumnos?**

A los estudiantes de postgrado se les asesora en los trámites que tengan que realizar durante el tiempo que dure el postgrado respecto a derechos y obligaciones que hay entre partes.

**12.- ¿Han tenido problemas con algún docente?**

Si, respecto al avance de materias.

**13.- ¿Creen que tienen la necesidad de algún tipo de sistema?**

Es conveniente tener un sistema de generación de una base de datos y archivos de estudiantes.

**14.- ¿Cuáles son los rangos de precios de los cursos que ofrecen?**

El rango de precios de cursos es de Bs. 500 a Bs. 1600.

Y el rango de postgrados es de:

-Diplomados Bs. 5000 a Bs. 7000

-Especialidades Bs. 9000 a Bs. 15000

-Maestrias Bs. 25000 a Bs. 45000

**15.- ¿Qué tipo de sistema les gustaría tener?**

Es conveniente tener un sistema de generación de una base de datos y archivos de estudiantes.

**16.- ¿La empresa cuenta con todos los permisos para su funcionamiento?**

Contamos con todos los permisos necesarios.

**19.- ¿Nos ayudaría a realizar un sistema de información para gestionar alumnos, docentes, cursos y ventas de la empresa?**

Nos gustaría ayudarlos.

1. asd

**ELEMENTOS DEL SISTEMA BASADO**

**EN COMPUTADORAS**

* 1. **Hardware**
     1. **Servidor**

**Configuraciones de hardware mínima**

* PC en optimas calidades
* Procesador Intel Core i3
* Memoria de 1 TB
* Ambiente en condiciones y temperaturas adecuadas
* Buena conexión de electricidad
* Velocidad de internet de 2 MB
  + 1. **Cliente**
* Procesador Intel Core i3
* Memoria de 1 TB
* Mouse
* Teclado
* Un monitor
* Velocidad de internet de 2 MB
  + 1. **Medios de comunicación**

**Materiales para instalar una Red**

* Conectores RJ45: Uno de los conectores principales utilizados con tarjetas

de red Ethernet transmite información a través de cables par trenzado.

* Cable (UTP) Categoría 5: Tipo de cableado más solicitado. Es un estándar

dentro de las comunicaciones de redes LAN.

* Grimpadora: Son una especie de alicates y sirven para fijar los cables.
* Tester de Cable: Sirve para medir magnitudes eléctricas en diferentes

ámbitos.

* Tarjeta de Red: Permite la comunicación entre diferentes aparatos

conectados entre sí.

* Hub o Switch: Llevan a cabo la conectividad de una red local (LAN). La

mayoría soportan cables cruzados.

* + 1. **Otros dispositivos**
* Parlantes
* Impresoras
* Memorias USB
  1. **Software**
     1. **Servidor**
* MySQL o SQL Server
* Procesador Intel Core i3 en adelante
* Lenguaje de programación PHP en PHPStorm
* Sistema operativo Windows 7 en adelante
  + 1. **Cliente**
* Sistema Operativo Windows 7 en adelante
* Navegador (Chrome, Mozilla, Opera, etc.)
* Lenguaje de programación PHP
  + 1. **Otro software adicional**
* Microsoft office Word
  1. **Datos**

El sistema de datos manejará lo siguientes datos:

* Informacion de todas las personas que tengan vinculo con INEGAS, es decir: Personal, Clientes y Docentes
* Estrucutra organizacional de INEGAS, como ser: cargos y departamentos.
* Organización de los cursos, se almacenaran los datos de los cursos que se da, los horarios, aulas y gestión.
  1. **Procesos**

**Registro:** En este proceso, se registran las personas. Toda persona que solicite información por los cursos por primera vez tiene que ser registrado y a esta persona se le pide sus datos personales. Los trabajadores que manejen este sistema de información deben tener registrado sus datos personales además de tener un nombre de usuario y contraseña para poder manejar el sistema.

**Inscripcion:** Se registrará las inscripciones de los alumnos, incluyendo la información del curso, el costo e información de la persona que le esta vendiendo.

**Cobranzas:** Se registrará los pagos que realicen los alumnos por cada cuota que se les asigno en la venta.

* 1. **Gente/Usuario**

El sistema tendrá en capacidad de tener en su dominio a todo el personal apto para realizar algún registro de cursos, personal o alumnos. También a las personas que realicen ventas a los alumnos.

* 1. **Documento**

La información que respalda al sistema es:

* Manual de usuario

**TECNOLOGIA PARA EL DESARROLLO**

**DEL SOFTWARE**

1. **Sdf**
   1. **Estrategia para el desarrollo del software**

Se utilizará el método del ciclo de vida básico, en este método deben cumplirse, sin pasarse por alto ni una de las siguientes etapas:

* Investigación o estudio preliminar.
* Determinación de los requerimientos de sistemas.
* Diseño del sistema.
* Desarrollo del software (Implementación).
* Prueba del Sistema.
* Implantación y evaluación del nuevo sistema (Mantenimiento).
  1. **Metodología para el desarrollo del software**
     1. **Características del PUDS**

El Proceso Unificado de Desarrollo de Software (PUDS) es un método de desarrollo de software compuesto por componentes e interconectados a través de interfaces bien definidas. El Proceso Unificado utiliza el lenguaje unificado de modelado para preparar todos los esquemas de un sistema de software.

El Proceso Unificado está dirigido por casos de uso, esto quiere decir que es el conjunto de casos de uso que describen la funcionalidad del sistema, donde un caso de uso es la representación de un requisito funcional dado por un usuario que proporciona un resultado importante.

La arquitectura del software surge de las necesidades de la empresa, como las que perciben los usuarios e inversores, y se refleja en los casos de uso. Sin embargo, también se ve influida por muchos otros factores como el sistema operativo, sistema gestor de base de datos, protocolos para comunicación de red, etc. Por un lado, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo. Por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, ahora y en el futuro. En realidad, tanto la arquitectura como los casos de uso deben evolucionar en paralelo.

La duración de desarrollo de un producto de software puede ir desde un par de meses hasta un año o más, es por eso que se divide el trabajo en mini-proyectos los cuales deben seleccionarse y ejecutarse de forma planificada. La importancia de la distribución del trabajo en partes más pequeñas es la minimización de pérdidas en casos de falla, es decir que, si algo malo pasa en el transcurso de desarrollo del software, esta falla afectará únicamente a una pequeña parte del trabajo total. El proceso unificado se repite a lo largo de 4 grandes fases: Inicio, elaboración, construcción y transición. Estos ciclos son los que describen al proceso desde su nacimiento hasta su muerte.

**Las cuatro “P”**

Para poder explicar mejor los conceptos del PUDS, se utilizan los términos Personas, Proyecto, Producto, Proceso y Herramientas.

**Personas**, se refiere a los seres humanos que entran en el ámbito del desarrollo de un software, como los arquitectos, desarrolladores, ingenieros de prueba y el personal de gestión que les da el soporte. El proceso que gestiona el desarrollo debe estar orientado a personas, es decir debe funcionar bien para las personas que lo utilicen.

**Proyecto**, es el elemento a través del cual se gestiona el desarrollo del software, donde un producto es el resultado de un proyecto.

**Producto**, son los artefactos que se crean durante la vida del proyecto, como los modelos, código fuentes, ejecutables y documentación.

**Proceso**, es un conjunto de actividades o pasos para convertir los requerimientos del usuario en producto, en otras palabras, un proceso es una plantilla para crear proyectos.

**Herramientas**, son los softwares que utilizamos para poder crear las actividades definidas en el proceso.

Un proceso dirigido por casos de uso, ya que los casos de uso proporcionan un medio sistemático e intuitivo de capturar requisitos funcionales y dirigen todo el proceso de desarrollo ya que la mayoría de las actividades se llevan a cabo partiendo de los casos de uso.

La importancia de la arquitectura de software empieza en la forma de administrar al personal en sus respectivos cargos de trabajo, es decir, cada quien debe realizar unas funciones específicas para que el resultado sea óptimo y estable. La descripción de la arquitectura permite al arquitecto controlar el desarrollo del sistema desde la perspectiva técnica. Se desarrolla de forma iterativa durante la fase de elaboración pasando por los requisitos, el análisis, el diseño, la implementación y las pruebas. Se describe como una vista de los modelos del sistema, de los modelos de casos de uso, análisis, diseño, implementación y despliegue.

* + 1. **Características del UML**

El Lenguaje Unificado de Modelado es un método de especificación, visualización, construcción y documentación de los artefactos de un sistema software. El objetivo más importante es que UML es un lenguaje de modelado de propósito general que pueden utilizar todos los modeladores. UML No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Uno de los objetivos obsoletos de UML era ser lo más simple y sencillo posible, es decir que sea algo pequeño y fácil de entender. Pero al haber un gran avance en la tecnología, esto se ha vuelto imposible, debido al tamaño de complejidad de los sistemas actuales en comparación a los antiguos.

Una de las razones más importantes para aplicar UML es producir software de una manera consistente que satisface las necesidades de sus usuarios. La importancia de modelar es simplemente la necesidad de planear algo, es decir, en el proceso de querer hacer algo o convertir algo, debemos saber cómo lo vamos a hacer, como lo vamos a hacer, etc.

El modelado sigue cuatro principios básicos:

*La elección acerca de qué modelos crear, tiene una*

*profunda influencia sobre cómo se acomete un problema y*

*como se da forma a una solución.*

*Todo modelo puede ser expresado con*

*diferentes niveles de precisión.*

*Los mejores modelos están ligados a la realidad.*

*Un único modelo o vista no es suficiente. Cualquier sistema no trivial se aborda mejor a través de un pequeño conjunto de modelos caso independientes con múltiples puntos de vista.*

* 1. **Herramientas de desarrollo**
     1. **Software**
* **Gestor de bases de datos SQL Server.** Este es el sistema gestor

de bases de datos que usaremos para la implementación de la base de datos para el sistema dándole prioridad por ser una de las mejores herramientas en esta área de desarrollo.

* **Lenguajes de programación PHP (con Framework Laravel).** Ofrece

muchas posibilidades para el desarrollo de software. Usaremos específicamente el entorno para implementar la interfaz del sistema, que es la que se usa para interactuar con el usuario.

* **Herramientas UML.** Entre las que usaremos está Enterprise Architect,

que es una herramienta UML que específicamente nos servirá para implementar los diferentes diagramas necesarios para el proceso de desarrollo del sistema,

* + 1. **Hardware**
* Procesador AMD A8
* Procesador Intel Inside
* Intel HD Graphics con memoria gráfica compartida.
* RAM 8 GB DDR3 (Maximo 16 GB).
* Disco duro 1 TB (5400 RPM) Serial ATA.
* Pantalla 16.0” Widescreen HD TruBrite resolución 1366 X 768 LED backlit
* Wi-Fi Wireless

1. **Posibles costos**

El costo del Hardware será de acuerdo a la empresa en la cual se implemente el sistema. A continuación, detallamos el costo de algunos equipos que se necesitaran.

**Equipo Costo Cantidad Uso**

PC Servidor 1300$us 1 Servidor central

PC/PC Portatil 1300$us 5 Para las oficinas o

departamentos

gerenciales.

Impresora 100$us 5 Impresión de

formularios

Internet 35$us 1 Para el servidor

central

1. **Posibles beneficios**
   1. **Tiempo**

* Aumento de la eficiencia en el proceso de registro y administración de los

estudiantes y los cursos que se ofrecen.

* Satisfacción por parte de alumnos y docentes por una mejor atención.
* Generar informes e indicadores, que permitan obtener información precisa

de los cursos dados y de los futuros cursos.

* Evitar pérdida de tiempo realizando cálculos mentales sobre cantidades de

alumnos inscritos en cada curso

* Reducir tiempo en realizar un registro de venta, entrega y cotización de los

cursos.

* 1. **Esfuerzo**
* Se reduce el esfuerzo por parte del personal al momento de registrar o

eliminar a un alumno de un curso.

* Existirá una mejor organización en el flujo de trabajo entre los empleados de

oficinas y los docentes.

* Se contará con detalles de los cursos que se ofrecen, lo cual, facilitara labúsqueda de los mismos para sus respectivas inscripciones.
  1. **Costos**

Con el sistema reducimos el costo de mano de obra, así como también el costo de material de escritorio para las funciones manuales que se utiliza cotidianamente.

1. **Posibles clientes**

* **OVICENTER:** Pequeña empresa dedicada a la venta de productos Tramontina, Crisa, Oxford, etc. Esta se encuentra ubicada en la Av. Pirai 2do anillo Calle Robore, Esq. Espejos Nº 05. Cuenta con un teléfono para la atención al cliente: 357 – 1817; y su página web www.ovicenter.com.bo.
* **Comerciantes:** Se pretende abastecer de estos productos los comerciantes de la feria Barrio Lindo y a los ferreteros que se encuentran detrás de la terminal.
* **Microempresarios:** Se pretende abastecer a los microempresarios que tienen puestos de venta de herramientas por toda la ciudad, manteniendo asi un constante crecimiento de la empresa.

**MODELO DE DOMINIO**

1. **Sd**
   1. **Diseño lógico**
      1. **Diagrama de clases**

****

* + 1. **Mapeo**

|  |  |
| --- | --- |
| LLAVE PRIMARIA | LLAVE FORANEA |

**Persona**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| codigo | ci | nombre | apellidoPaterno | apellidoMaterno |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| fechaNacimiento | telefono | correo | tipoAlumno | tipoDocente | tipoTrabajador |

**Alumno**

|  |  |  |
| --- | --- | --- |
| codigo | codigoPersona | profesion |

**Usuario**

|  |  |  |  |
| --- | --- | --- | --- |
| nombre | contraseña | privilegio | idPrivilegio |

**Departamento**

|  |  |  |
| --- | --- | --- |
| Id | nombre | cantidadCargos |

**Cargo**

|  |  |  |  |
| --- | --- | --- | --- |
| id | nombre | estado | idDepartamento |

**PersonalAdministrativo**

|  |  |  |  |
| --- | --- | --- | --- |
| codigo | codigoPersona | añosExperiencia | idCargo |

**IntervaloDePago**

|  |  |
| --- | --- |
| id | nombre |

**MetodoDePago**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | nombre | plazo | idIntervaloDePago |

**Docente**

|  |  |  |
| --- | --- | --- |
| codigo | codigoPersona | profesión |

**Tipo**

|  |  |
| --- | --- |
| Id | nombre |

**Docente\_Tipo**

|  |  |
| --- | --- |
| codigoDocente | idTipo |

**Curso**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| codigo | nombre | duracion | monto | capacidad | idTipo |

**NotaDeVenta**

|  |  |  |  |
| --- | --- | --- | --- |
| nro | fecha | codigoPersonalAdministrativo | codigoAlumno |

|  |  |
| --- | --- |
| idMetodoDePago | codigoCurso |

**Cuota**

|  |  |  |  |
| --- | --- | --- | --- |
| nroNotaDeVenta | nro | Monto | estado |

**PagoDeCuota**

|  |  |  |  |
| --- | --- | --- | --- |
| nro | fecha | nroNotaDeVenta | nroCuota |

**Aula**

|  |  |  |
| --- | --- | --- |
| nro | ubicacion | capacidad |

**Horario**

|  |  |  |
| --- | --- | --- |
| Id | horaInicio | horaFin |

**Gestion**

|  |  |
| --- | --- |
| nro | nombre |

**Grupo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | nombre | nroAula | codigoCurso | idHorario | nroGestion | codigoDocente |

* + 1. **Normalización**

**Primera Forma Normal**

**Persona**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Código | ci | nombre | apellidoPaterno | apellidoMaterno |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| fechaNacimiento | correo | tipoAlumno | tipoDocente | tipoTrabajador |

**Telefono**

|  |  |
| --- | --- |
| codigoPersona | Numero |

**Primera Forma Normal**

**Docente**

|  |  |
| --- | --- |
| codigo | codigoPersona |

**Profesion**

|  |  |
| --- | --- |
| codigoDocente | nombre |

* 1. **Diseño físico**
     1. **Tabla de volumen**

**Persona**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| codigo | Numérico entero | Identificador | 5 bytes | No | Primaria |
| Ci | Numérico entero | Carnet de identidad | 9 bytes | No | NO |
| nombre | Cadena | Nombre de la persona | 30 bytes | No | NO |
| apellidoPaterno | Cadena | Apellido de la persona | 40 bytes | No | NO |
| apellidoMaterno | Cadena | Apellido de la persona | 40 bytes | No | NO |
| telefono | Numerico entero | Numero de Teléfono | 8 bytes | Si | NO |
| Correo | Cadena | Correo electronico | 50 bytes | Si | NO |
| fechaNacimiento | Cadena | Fecha de nacimiento | 10 bytes | No | NO |
| tipoAlumno | booleano | Verificador que es un alumno | 1 bit | No | NO |
| tipoDocente | booleano | Verificador que es un docente | 1 bit | No | NO |
| tipoTrabajador | booleano | Verificador que es un trabajador | 1 bit | No | NO |

**Alumno**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| codigo | Numérico entero | Identificador como alumno | 5 bytes | No | Primaria |
| codigoPersona | Numérico entero | Código de la persona que será registrada como alumno | 5 bytes | No | Foranea |
| profesion | Cadena | Profesión del alumno | 40 bytes | Si | NO |

**Usuario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| nombre | Cadena | Usuario de la persona | 30 bytes | No | Primaria |
| contraseña | Cadena | Contraseña del usuario | 30 bytes | No | NO |
| codigoPersona | Numérico entero | Código de la persona del usuario | 5 bytes | No | Foranea |
| nivelPrivilegio | Numerico entero | Nivel de acceso del usuario | 2 bytes | No | NO |

**Departamento**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| Id | Numérico entero | Identificador como departamento | 5 bytes | No | Primaria |
| nombre | Cadena | Nombre del departamento | 50 bytes | No | NO |
| cantidadCargos | Numerico entero | Cantidad de cargos en el depto. | 2 bytes | No | NO |

**Cargo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| Id | Numérico entero | Identificador como cargo | 5 bytes | No | Primaria |
| nombre | cadena | Nombre del cargo | 30 bytes | No | NO |
| estado | booleano | Verifica si el estado esta usado | 1 bit | No | NO |
| idDepartamento | Numérico entero | Numero del departamento al que pertence | 5 bytes | No | Foranea |

**PersonalAdministrativo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| codigo | Numérico entero | Identificador como trabajador | 5 bytes | No | Primaria |
| codigoPersona | Numérico entero | Código de la persona que será registrada como trabajador | 5 bytes | No | Foranea |
| añosExperiencia | Numérico entero | Años de experiencia del trabajador | 2 bytes | Si | NO |
| idCargo | Numerico entero | Identificador del cargo del trabajador | 5 bytes | No | Foranea |

**IntevaloDePago**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | Identificador como intervalo de pago | 5 bytes | No | Primaria |
| nombre | cadena | Nombre del intervalo de pago | 40 bytes | No | NO |

**MetodoDePago**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | Identificador como metodo de pago | 5 bytes | No | Primaria |
| nombre | cadena | Nombre del metodo de pago | 40 bytes | No | NO |

**Docente**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| codigo | Numérico entero | Código del docente | 5 bytes | No | Primaria |
| codigoPersona | cadena | Nombre del intervalo de pago | 40 bytes | No | Foranea |
| profesion | cadena | Profesión del docente | 40 bytes | No | NO |

**Tipo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | Identificador como tipo | 5 bytes | No | Primaria |
| Nombre | cadena | Nombre del tipo | 30 bytes | No | NO |

**Docente\_Tipo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| codigoDocente | Numérico entero | Código del docente | 5 bytes | No | Primaria  Foranea |
| idTipo | Numérico entero | Identificador del tipo | 5 bytes | No | Primaria  Foranea |

**Curso**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| codigo | Numérico entero | Código del curso | 5 bytes | No | Primaria |
| nombre | Cadena | Nombre del curso | 30 bytes | No | NO |
| duracion | Numérico entero | Duración del curso en meses | 2 bytes | No | NO |
| monto | Numérico fraccional | Costo del curso | 10 bytes | No | NO |
| capacidad | Numérico entero | La cantidad de alumnos que entran en el curso | 2 bytes | No | NO |
| idTipo | Numérico entero | Tipo del curso | 5 bytes | No | foranea |

**NotaDeVenta**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| nro | Numérico entero | Numero de la nota de venta | 5 bytes | No | Primaria |
| fecha | Numérico entero | Fecha en la que se realiza la nota de venta | 10 bytes | No | NO |
| monto | Numérico fraccionario | Monto total a pagar | 10 bytes | No | NO |
| codigoPersonal  Administrativo | Numérico entero | Código de la persona que realiza la venta | 5 bytes | No | Foranea |
| codigoAlumno | Numérico entero | Código del alumno a quien se vende | 5 bytes | No | Foranea |
| idMetodoPago | Numérico entero | Identificador del método de pago | 5 bytes | No | Foranea |
| codigoCurso | Numérico entero | Código del curso por el que se va a pagar | 5 bytes | No | Foranea |

**Cuota**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| nroNotaDeVenta | Numérico entero | Numero de la nota de venta que se va a pagar | 5 bytes | No | Primaria  Foranea |
| Nro | Numérico entero | Numero de la cuota | 5 bytes | No | Primaria |
| Monto | Numérico flotante | Monto que se paga en cada cuota | 10 bytes | No | NO |
| estado | booleano | Verifica si la cuota ya esta pagada | 1 bit | No | NO |

**PagoDeCuota**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| Nro | Numérico entero | Numero de pago de cuota | 5 bytes | No | Primaria |
| Fecha | Date | Fecha en la que se realiza el pago | 10 bytes | No | NO |
| nroNotaDeVenta | Numérico entero | Numero de la nota de venta a la que pertenecera el pago | 5 bytes | No | Foranea |
| nroCuota | Numérico entero | Numero de la cuota que se esta pagando | 5 bytes | No | Foranea |

**Aula**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| Nro | Numérico entero | Numero del aula | 5 bytes | No | Primaria |
| ubicacion | cadena | Descripcion de la ubicación del aula | 30 bytes | No | NO |
| Capacidad | Numérico entero | Capacidad máxima del aula | 3 bytes | No | NO |

**Horario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| id | Numérico entero | Identificador del horario | 5 bytes | No | Primaria |
| horaInicio | cadena | Hora de inicio del horario | 5 bytes | No | NO |
| horaFin | cadena | Hora de fin del horario | 3 bytes | No | NO |

**Gestion**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| Nro | Numérico entero | Numero de la gestión | 5 bytes | No | Primaria |
| Nombre | Cadena | Nombre de la gestion | 30 bytes | No | NO |

**Grupo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Descripción** | **Tamaño** | **Nulo** | **Llave** |
| Id | Numérico entero | Identificador del horario | 5 bytes | No | Primaria |
| Nombre | cadena | Hora de inicio del horario | 5 bytes | No | NO |
| nroAula | Numérico entero | Numero del aula del grupo | 5 bytes | No | Foranea |
| codigoCurso | Numérico entero | Código del curso del grupo | 5 bytes | No | Foranea |
| idHorario | Numérico entero | Identificador del horario del curso | 5 bytes | No | Foranea |
| nroGestion | Numérico entero | Numero de la gestión en la que se da el curso | 5 bytes | No | Foranea |
| codigoDocente | Numérico entero | Código del docente que da el curso en el grupo | 5 bytes | No | Foranea |

* + 1. **Script**

create database SIinegas;

use SIinegas;

create table persona(

codigo int primary key,

ci int not null,

nombre varchar(30) not null,

apellidoPaterno varchar(40) not null,

apellidoMaterno varchar(40) not null,

telefono int,

correo varchar(50),

fechaNacimiento date not null,

tipoAlumno bit not null,

tipoDocente bit not null,

tipoTrabajador bit not null

);

create table alumno(

codigo int primary key,

codigoPersona int not null,

profesion varchar(40),

foreign key(codigoPersona)references persona(codigo)

);

create table usuario(

nombre varchar(30) not null,

contraseña varchar(30) not null,

privilegio int not null,

codigoPersona int not null,

foreign key(codigoPersona)references persona(codigo)

);

create table departamento(

id int primary key,

nombre varchar(50) not null,

cantidadCargos int not null

);

create table cargo(

id int primary key,

nombre varchar(30) not null,

estado bit not null,

idDepartamento int not null,

foreign key (idDepartamento) references departamento (id)

on update cascade

on delete cascade

);

create table personalAdministrativo(

codigo int primary key,

codigoPersona int not null,

añosExperiencia int,

idCargo int not null,

foreign key (codigoPersona) references persona (codigo),

foreign key (idCargo) references cargo (id)

);

create table intervaloDePago(

id int primary key,

nombre varchar(40) not null

);

create table metodoDePago(

id int primary key,

nombre varchar(40) not null,

plazo int not null,

idIntervaloDePago int not null,

foreign key(idIntervaloDePago) references intervaloDePago(id)

);

create table docente(

codigo int primary key,

codigoPersona int,

profesion varchar(40) not null,

foreign key(codigoPersona)references persona(codigo)

);

create table tipo(

id int primary key,

nombre varchar(30) not null

);

create table docente\_tipo(

codigoDocente int,

idTipo int,

primary key (codigoDocente, idTipo),

foreign key (codigoDocente) references docente(codigo),

foreign key (idTipo) references tipo (id)

);

create table curso(

codigo int primary key,

nombre varchar(30) not null,

duracion int not null,

monto float not null,

capacidad int not null,

idTipo int not null,

foreign key (idTipo) references tipo (id)

);

create table notaDeVenta(

nro int primary key,

fecha date not null,

monto float not null,

codigoPersonalAdministrativo int not null,

codigoAlumno int not null,

idMetodoPago int not null,

codigoCurso int not null,

foreign key (codigoPersonalAdministrativo) references personalAdministrativo(codigo),

foreign key (codigoAlumno) references alumno (codigo),

foreign key (idMetodoPago) references metodoDePago (id),

foreign key (codigoCurso) references curso (codigo)

);

create table cuota(

nroNotaDeVenta int,

nro int,

monto float not null,

estado bit not null,

primary key(nroNotaDeVenta,nro),

foreign key(nroNotaDeVenta) references notaDeVenta(nro)

);

create table pagoDeCuota(

nro int primary key,

fecha date not null,

nroNotaDeVenta int not null,

nroCuota int not null,

foreign key (nroNotaDeVenta,nroCuota) references cuota(nroNotaDeVenta,nro)

);

create table aula(

nro int primary key,

ubicacion varchar(30)not null,

capacidad int not null

);

create table horario(

id int primary key,

horaInicio varchar(5) not null,

horaFin varchar(5) not null

);

create table gestion(

nro int primary key,

nombre varchar(30) not null

);

create table grupo(

id int primary key,

nombre varchar(30) not null,

nroAula int not null,

codigoCurso int not null,

idHorario int not null,

nroGestion int not null,

codigoDocente int not null,

foreign key(nroAula)references aula(nro),

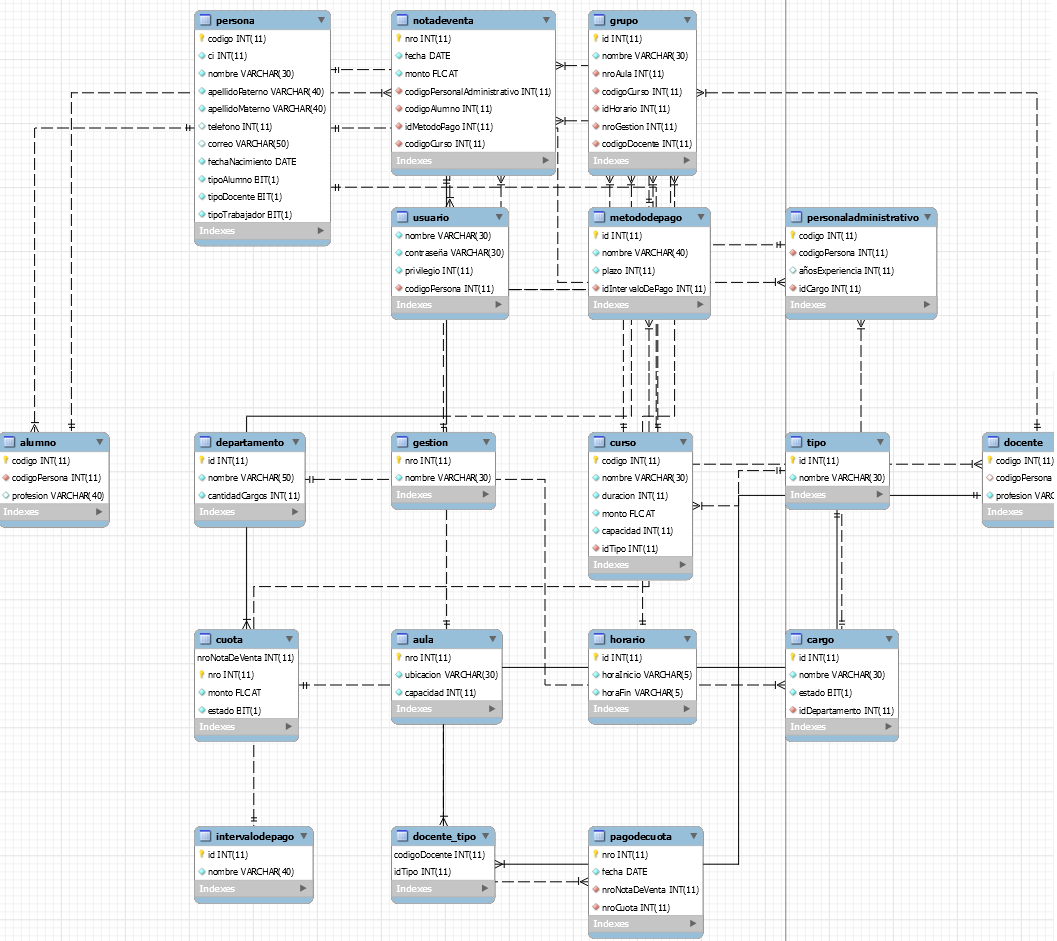
foreign key(codigoCurso)references curso(codigo),

foreign key(idHorario)references horario(id),

foreign key(nroGestion)references gestion(nro),

foreign key(codigoDocente)references docente(codigo)

);

* + 1. **Diagrama relacional**
    2. **Actualización de tuplas**

insert into persona values (1,111,'Abigail','Gutierrez','Justiniano',72520129,'abigutierrez@gmail.com','1995-02-12',1,0,0);

insert into persona values (2,222,'Sebastian','Alvarez','Roca',72520129,'sebasalvarez@gmail.com','1985-08-26',1,0,0);

insert into persona values (3,333,'Pablo','Ricaldi','Moron',72520129,'pabloricaldi@gmail.com','1988-10-12',1,0,0);

insert into persona values (4,444,'Luisa','Gallardo','Suarez',72520129,'luisagallardo@gmail.com','1992-04-12',1,0,0);

insert into persona values (5,555,'Carlos','Sandoval','Padilla',72520129,'carlossandoval@gmail.com','1996-07-12',0,1,0);

insert into persona values (6,666,'Juan','Zegarra','Lopez',72520129,'juanzegarra@gmail.com','1991-06-11',0,1,0);

insert into persona values (7,777,'Norma','Soliz','Lujan',72520129,'normasoliz@gmail.com','1987-12-22',0,0,1);

insert into persona values (8,888,'Karen','Perez','Herrera',76385791,'karenperez@gmail.com','1995-02-18',0,0,1);

insert into persona values (9,999,'Simon','Aguilar','Duran',79230092,'simonaguilar@gmail.com','1995-02-18',0,0,1)

insert into alumno values (1,1,'Ingeniero Petrolero');

insert into alumno values (2,2,'Ingeniero Quimico');

insert into alumno values (3,3,null);

insert into alumno values (4,4,'Ingeniero Civil');

insert into usuario values ('Abigail','abigail123',10,1);

insert into usuario values ('Sebastian','sebastian123',10,2);

insert into usuario values ('Luisa','luisa123',10,4);

insert into usuario values ('Juan','juan123',10,6);

insert into usuario values ('Norma','norma123',10,7);

insert into usuario values ('Karen','karen123',10,8);

insert into usuario values ('Simon','simon123',10,9);

insert into departamento values (1,'Departamento de ventas',0);

insert into departamento values (2,'Departamento de marketing',0);

insert into departamento values (3,'Departamento de R.R.H.H.',0);

insert into cargo values (1,'Ejecutivo de ventas',1,1);

insert into cargo values (4,'Director de publicidad',1,2);

insert into cargo values (3,'Director de ventas',1,2);

insert into cargo values (2,'Desempleado',0,3);

insert into personalAdministrativo values (1,7,2,1);

insert into personalAdministrativo values (2,8,4,3);

insert into personalAdministrativo values (3,9,3,2);

insert into intervaloDePago values (1,'Mensual');

insert into intervaloDePago values (2,'Bimestral');

insert into intervaloDePago values (3,'Trimestral');

insert into intervaloDePago values (6,'Semestral');

insert into intervaloDePago values (10,'Una vez');

insert into metodoDePago values (1,'Rapido',1,2);

insert into metodoDePago values (2,'Medio',2,2);

insert into metodoDePago values (3,'Lento',3,2);

/\*LAS CUOTAS SE INSERTAN AUTOMATICAMENTE CON UN TRIGGER\*/

insert into docente values (1,5,'Ingeniero Petrolero');

insert into docente values (2,6,'Ingeniero Petrolero');

insert into tipo values (1,'Continuo');

insert into tipo values (2,'Postgrado');

insert into docente\_tipo values (1,2);

insert into docente\_tipo values (2,1);

insert into docente\_tipo values (2,2);

insert into curso values (1,'Excel Basico',3,2500,25,1);

insert into curso values (2,'Word Avanzado',1,2000,20,1);

insert into curso values (3,'Maestria en Procesos',2,2000,25,2);

insert into curso values (4,'Maestria en escabacion',3,3000,25,2)

insert into notaDeventa values (1,'2018-01-26',2500,1,1,1,1);

insert into notaDeventa values (2,'2018-01-26',2000,1,2,2,2);

insert into notaDeventa values (3,'2018-01-26',2000,1,3,3,3);

insert into notaDeventa values (4,'2018-01-26',3000,1,4,1,4);

insert into aula values (1,'Primer piso',25);

insert into aula values (2,'Segundo piso',20);

insert into aula values (3,'Segundo piso',25);

insert into aula values (4,'Tercer piso',25);

insert into horario values (1,'10:00','12:00');

insert into horario values (2,'15:00','18:00');

insert into horario values (3,'20:00','22:00');

insert into gestion values (1,'1-2018');

insert into gestion values (2,'2-2018');

insert into grupo values (1,'Mañana',1,1,1,1,1);

insert into grupo values (2,'Tarde',2,3,2,1,2);

insert into grupo values (3,'Noche',3,2,3,2,1);

insert into grupo values (4,'Mañana',4,4,1,2,2);

insert into grupo values (5,'Tarde',2,1,2,1,2);

update persona set nombre='Carmen' where ci=888;

update usuario set nombre='Carmen' where codigoPersona=8;

update docente set profesion='Ingeniero Industrial' where codigoPersona=5;

update horario set horaInicio='15:00' where id=1;

update horario set horaFin='17:00' where id=1;

update curso set duracion=1 where codigo=4;

update notaDeVenta set fecha='2018-01-20' where nro=3;

delete from usuario where nombre='Norma';

delete from usuario where nombre='Simon';

delete from aula where nro=3

* + 1. **Consultas**

/\*1 Mostrar los trabajadores del

departamento de ventas\*/

select persona.codigo,persona.nombre,apellidoPaterno,apellidoMaterno

from departamento,cargo,personalAdministrativo,persona

where persona.codigo=personalAdministrativo.codigoPersona and

personalAdministrativo.idCargo=cargo.id and

cargo.idDepartamento=departamento.id and

departamento.nombre='Departamento de ventas'

/\*2 Mostrar los docentes que dan cursos de postGrado\*/

select persona.nombre,persona.apellidoPaterno,persona.apellidoMaterno

from docente,tipo,docente\_tipo,persona

where persona.codigo=docente.codigoPersona and

docente\_tipo.codigoDocente=docente.codigo and

docente\_tipo.idTipo=tipo.id and

tipo.nombre='PostGrado'

/\*3 Mostrar los cursos del alumno Juan Perez\*/

select curso.codigo,curso.nombre

from curso,notaVenta,alumno,persona

where alumno.codigo=notaVenta.codigoAlumno and

notaVenta.codigoCurso=curso.codigo and

alumno.codigoPersona=persona.codigo and

persona.nombre='Juan' and

persona.apellidoPaterno='Perez'

/\*4 Mostrar a los trabajadores que realizaron ventas el 2019-02-02\*/

select persona.nombre,persona.apellidoPaterno,persona.apellidoMaterno

from personalAdministrativo,notaVenta,persona

where notaVenta.codigoPersonalAdministrativo=personalAdministrativo.codigo and

persona.codigo=personalAdministrativo.codigoPersona and

notaVenta.fecha='2019-02-02'

/\*5 Mostrar a los docentes y sus usuarios\*/

select persona.nombre,usuario.nombre

from persona,usuario,personalAdministrativo

where persona.codigo=personalAdministrativo.codigoPersona and

usuario.codigoPersona=persona.codigo

/\*6 Mostrar en cuantos grupos se da el curso Excel basico\*/

select count(\*)

from grupo,curso

where curso.codigo=grupo.codigoCurso and

curso.nombre='Excel basico'

/\*7 Mostrar la cantidad de aulas usadas para todos

los cursos de postGrado\*/

select count(\*)

from aula,grupo,curso

where grupo.codigoCurso=curso.codigo and

aula.nro=grupo.nroAula and

curso.idTipo = tipo.id and

tipo.nombre='postGrado'

/\*8 Mostrar todas las ventas que se hicieron

con un intervalo de pago Mensual\*/

select notaVenta.nro,notaVenta.fecha

from notaVenta,metodoDePago,intervaloDePago

where notaVenta.idMetodoPago=metodoDePago.id and

metodoDePago.idIntervaloDePago=intervaloDePago.id and

intervaloDePago.nombre='Mensual'

/\*9 Mostrar las aulas en las que Juan Perez pasa clases\*/

select aula.nro,aula.ubicacion

from persona,alumno,notaVenta,curso,grupo,aula

where persona.codigo=alumno.codigoPersona and

alumno.codigo=notaVenta.codigoAlumno and

curso.codigo=notaVenta.codigoCurso and

grupo.codigoCurso=curso.codigo and

aula.nro=grupo.nroAula

/\*10 Mostrar la cantidad de cuotas por pagar

de la nota de venta 3\*/

select count(\*)

from cuota,metodoDePago,notaVenta

where notaVenta.idMetodoPago=metodoDePago.id and

metodoDePago.id=cuota.idMetodoDePago and

notaVenta.nro=3 and cuota.estado=0

/\*11Mostrar los cursos en los que se inscribieron

mas de 10 personas\*/

select curso.codigo,curso.nombre

from curso,notaVenta

where notaVenta.codigoCurso=curso.codigo

group by count(\*)

having count(\*)>10

/\*12Mostrar los cursos pagados por cada alumno\*/

select persona.nombre,persona.apellidoPaterno,curso.nombre

from alumno,notaVenta,persona,curso

where persona.codigo=alumno.codigoPersona and

alumno.codigo=notaVenta.codigoAlumno and

notaVenta.codigoCurso=curso.codigo

/\*13Mostrar la cantidad de inscritos en cada curso\*/

select curso.nombre, count(\*) as inscritos

from notaDeVenta,curso

where notaDeVenta.codigoCurso=curso.codigo

group by curso.nombre

/\*14Mostrar la cantidad de grupos en las que dará clases

un docente por la tarde\*/

select persona.nombre,count(\*)as cantidadGrupos

from grupo,docente,persona

where docente.codigo=grupo.codigoDocente and

docente.codigoPersona=persona.codigo and

grupo.nombre='Tarde'

group by persona.nombre

/\*15Mostrar las gestiones en las que el docente Juan dio cursos

de postgrado\*/

select gestion.nro,gestion.nombre

from gestion,grupo,docente,tipo,docente\_tipo,persona

where gestion.nro=grupo.nroGestion and

docente.codigo=grupo.codigoDocente and

docente.codigo=docente\_tipo.codigoDocente and

docente\_tipo.idTipo=tipo.id and

docente.codigoPersona=persona.codigo and

persona.nombre='Juan' and tipo.nombre='Postgrado'

/\*16Mostrar los alumnos que esten inscritos

en cursos Continuos\*/

select persona.nombre

from persona,alumno,notaDeVenta,curso,tipo

where persona.codigo=alumno.codigoPersona and

alumno.codigo=notaDeVenta.codigoAlumno and

notaDeVenta.codigoCurso=curso.codigo and

curso.idTipo=tipo.id and tipo.nombre='Continuo'

/\*17Mostrar los personales administrativos ordenados por el apellido que

no tienen usuario(nuevos) y trabajan en el departamento de ventas\*/

select persona.nombre,persona.apellidoPaterno

from persona,personalAdministrativo,usuario,cargo,departamento

where persona.codigo=personalAdministrativo.codigo and

persona.codigo=usuario.codigoPersona and cargo.id=personalAdministrativo.idCargo

and cargo.idDepartamento=departamento.id and departamento.nombre='Departamento de Ventas'

order by persona.apellidoPaterno asc

/\*18Mostrar la cantidad de alumnos que hay en cada aula\*/

select aula.nro,aula.ubicacion,aula.capacidad, count(\*)as cantidadAlumnos

from persona,alumno,notaDeVenta,curso,grupo,aula

where persona.codigo=alumno.codigoPersona and alumno.codigo=notaDeVenta.codigoAlumno

and notaDeVenta.codigoCurso=curso.codigo and curso.codigo=grupo.codigoCurso

and grupo.nroAula=aula.nro

group by aula.nro,aula.ubicacion,aula.capacidad

/\*19Mostrar los cursos con sus respectivas aulas en las que se dará clases en la tarde\*/

select curso.nombre,aula.nro,aula.ubicacion

from curso,grupo,aula

where curso.codigo=grupo.codigoCurso and grupo.nroAula=aula.nro and grupo.nombre='Tarde'

/\*20Mostrar los personales aministrativos que trabajan en el departamento de ventas

que tengan mas de 30 años\*/

select persona.nombre,persona.apellidoPaterno,persona.apellidoMaterno

from persona,personalAdministrativo,cargo,departamento

where persona.codigo=personalAdministrativo.codigoPersona and personalAdministrativo.idCargo=cargo.id

and cargo.idDepartamento=departamento.id and departamento.nombre='Departamento de venta' and

year(getdate())-year(persona.fechaNacimiento)>30

/\*21Monto total ganado por cada curso\*/

select curso.codigo,curso.nombre,sum(notaDeVenta.monto)as total

from notaDeVenta,curso

where notaDeVenta.codigoCurso=curso.codigo

group by curso.codigo,curso.nombre

/\*22Mostrar monto pagado de cada nota de venta hasta la fecha\*/

select notaDeVenta.nro,notaDeVenta.fecha,sum(cuota.monto)as totalPagado

from notaDeVenta,cuota,pagoDeCuota

where notaDeVenta.nro=cuota.nroNotaDeVenta and cuota.nro=pagoDeCuota.nroCuota

and getdate()>=pagoDeCuota.fecha

group by notaDeVenta.nro,notaDeVenta.fecha

/\*23Mostrar los cursos que se vendieron en la fecha 2018-01-26 con

método de pago Rapido\*/

select curso.codigo,curso.nombre

from notaDeVenta,curso,metodoDePago

where notaDeVenta.codigoCurso=curso.codigo and notaDeVenta.idMetodoPago=metodoDePago.id

and notaDeVenta.fecha='2018-01-26' and metodoDePago.nombre='Rapido'

/\*24Mostrar los horarios y aula de cada curso\*/

select curso.nombre,horario.horaInicio,horario.horaFin,aula.nro

from horario,grupo,curso,aula

where horario.id=grupo.idHorario and grupo.codigocurso=curso.codigo

and aula.nro=grupo.nroAula

/\*25Mostrar a los alumnos que terminaron de pagar todas sus cuotas\*/

select persona.nombre,persona.apellidoPaterno,persona.ci

from alumno,notaDeVenta,cuota,persona

where persona.codigo=alumno.codigoPersona and

notaDeVenta.codigoAlumno=alumno.codigo and

cuota.nroNotaDeVenta not in(

select nroNotaDeVenta

from cuota

where cuota.estado=0)

/\*26Mostrar todos los cursos que se dieron por gestion\*/

select gestion.nombre,curso.codigo,curso.nombre

from gestion,curso,grupo

where gestion.nro=grupo.nroGestion and

grupo.codigoCurso=curso.codigo

/\*27Mostrar a los alumnos que pagaron en efectivo\*/

select alumno.codigo,persona.nombre,persona.apellidoPaterno

from alumno,persona,notaDeVenta,metodoDePago

where persona.codigo=alumno.codigoPersona and

notaDeVenta.codigoAlumno=alumno.codigo and

notaDeVenta.idMetodoPago=metodoDePago.id and

metodoDePago.idIntervaloDePago=10

/\*28Mostrar a las personas que son docentes y alumnos\*/

select nombre,apellidoPaterno,ci

from persona,alumno,docente

where persona.codigo=alumno.codigoPersona and

persona.codigo=docente.codigoPersona

/\*29Mostrar los horarios disponibles de cada aula\*/

select aula.nro,aula.ubicacion,horario.horaInicio,horario.horaFin

from aula,horario,grupo

where aula.nro=grupo.nroAula and

grupo.idHorario<>horario.id

/\*30Mostrar la cantidad de ventas de cursos

de postgrado en el mes actual\*/

select count(\*)

from alumno,notaDeVenta,curso,tipo

where alumno.codigo=notaDeVenta.codigoAlumno and

notaDeVenta.codigoCurso=curso.codigo and

curso.idTipo=2 and

month(notaDeVenta.fecha)=month(getdate())

* + 1. **Procedimientos almacenados**

/\*1 Obtener la cantidad de cupos disponibles

para un curso\*/

create function cuposDisponibles(@codigoCurso int)returns int

as begin

declare @cuposVendidos int

set @cuposVendidos= (

select count(\*)

from notaDeVenta,curso

where codigoCurso=codigo and

codigo=@codigoCurso)

declare @cuposDelCurso int

set @cuposDelCurso = (

select capacidad

from curso

where codigo=@codigoCurso)

declare @cuposDisponibles int

set @cuposDisponibles=@cuposDelCurso-@cuposVendidos;

return @cuposDisponibles

end

/\*2 Obtener el monto que falta por pagar

de una nota de venta\*/

use SIinegas;

create function montoPorPagar(@nroNotaDeVenta int)returns float

as begin

declare @cantidadDeCuotasNoPagadas int

set @cantidadDeCuotasNoPagadas=(

select count(\*)

from cuota

where cuota.nroNotaDeVenta=@nroNotaDeVenta and

cuota.estado=0)

declare @montoFaltante float

set @montoFaltante=@cantidadDeCuotasNoPagadas\*(

select monto

from cuota

where nroNotaDeVenta=@nroNotaDeVenta)

return @montoFaltante

end

/\*3 Obtener la cantidad de cuotas que falta

por pagar de una nota de venta\*/

use SIinegas;

create function cantidadDeCuotasPorPagar(@nroNotaDeVenta int)returns int

as begin

declare @cantidadDeCuotasNoPagadas int

set @cantidadDeCuotasNoPagadas=(

select count(\*)

from cuota

where cuota.nroNotaDeVenta=@nroNotaDeVenta and

cuota.estado=0)

declare @montoFaltante float

set @montoFaltante=@cantidadDeCuotasNoPagadas\*(

select monto

from cuota

where nroNotaDeVenta=@nroNotaDeVenta)

return @montoFaltante

end

/\*4 Obtener el monto ganado

en un determinado mes\*/

create function montoGanado (@nroMes int)returns float

as begin

declare @nroPagoDeCuota int

declare @montoGanado float

set @montoGanado=0

declare cursorNroPagoDeCuota cursor for(

select nro

from pagoDeCuota

where month(fecha)=@nroMes)

open cursorNroPagoDeCuota

fetch from cursorNroPagoDeCuota into @nroPagoDeCuota

while(@@FETCH\_STATUS=0)

begin

set @montoGanado=@montoGanado+(

select monto

from cuota,pagoDeCuota

where pagoDeCuota.nroCuota=cuota.nro and

pagoDeCuota.nro=@nroPagoDeCuota)

fetch from cursorNroPagoDeCuota into @nroPagoDeCuota

end

close cursorNroPagoDeCuot

deallocate cursorNroPagoDeCuota

return @montoGanado

end

/\*5 Ver la lista de inscritos en un curso\*/

create function listaDeInscritos(@nombreCurso varchar(30))

returns table

as

return (select persona.codigo, persona.ci,persona.nombre,persona.apellidoPaterno,persona.apellidoMaterno

from persona,alumno,notaDeVenta,curso

where persona.codigo=alumno.codigoPersona and alumno.codigo=notaDeVenta.codigoAlumno

and curso.codigo=notaDeVenta.codigoCurso and curso.nombre=@nombreCurso)

/\*6 Ver los vendedores que vendieron mas de dos veces en el

mismo dia en algun mes\*/

create function vendedoresQueVendieronMasDeDosVeces(@mes int)

returns table

as

return (select persona.codigo,persona.ci,persona.nombre,count(\*)as cantidadVentas

from persona,departamento,cargo,personalAdministrativo,notaDeVenta

where departamento.id=cargo.idDepartamento and cargo.id=personalAdministrativo.idCargo

and personalAdministrativo.codigo=notaDeVenta.codigoPersonalAdministrativo and

persona.codigo=personalAdministrativo.codigoPersona and

month(notaDeVenta.fecha)=@mes and departamento.nombre='Departamento de ventas'

group by persona.codigo,persona.ci,persona.nombre

having count(\*)>2)

* + 1. **Disparadores**

/\*1 Verificar que el alumno a registrar

sea de tipo alumno\*/

create trigger verificarTipoAlumno

on alumno for insert as

if((select persona.tipoAlumno

from persona,inserted

where inserted.codigoPersona=persona.codigo)=0)

begin

print 'No es de tipo alumno'

rollback tran

end;

/\*2 Verificar que el docente a registrar

sea de tipo docente\*/

create trigger verificarTipoDocente

on docente for insert as

if((select persona.tipoDocente

from persona,inserted

where inserted.codigoPersona=persona.codigo)=0)

begin

print 'No es de tipo docente'

rollback tran

end

/\*3 Verificar que el trabajador a registrar

sea de tipo trabajador\*/

create trigger verificarTipoTrabajador

on personalAdministrativo for insert as

if((select persona.tipoTrabajador

from persona,inserted

where inserted.codigoPersona=persona.codigo)=0)

begin

print 'No es de tipo trabajador'

rollback tran

end

/\*4 Aumentar la cantidad de cargos en un departamento

al añadir un nuevo cargo\*/

create trigger cantidadDepartamento

on cargo for insert as

declare @idDpto int

set @idDpto=(select idDepartamento from inserted)

update departamento

set cantidadCargos=cantidadCargos+1

where departamento.id=@idDpto

/\*5 Disminuir la cantidad de cargos en un departamento

al eliminar un cargo\*/

create trigger cantidadDepartamentoB

on cargo for delete as

declare @idDpto int

declare idDeptoCursor cursor for (select idDepartamento from deleted)

open idDeptoCursor

fetch from idDeptoCursor into @idDpto

while(@@FETCH\_STATUS=0)

begin

update departamento

set cantidadCargos=cantidadCargos-1

where departamento.id=@idDpto

fetch from idDeptoCursor into @idDpto

end

close idDeptoCursor

/\*6 Verificar que la cantidad de cargos incial

al crear un departamento sea 0\*/

create trigger cantidadIncialDepartamento

on departamento for insert as

if((select cantidadCargos

from inserted)!=0)

begin

print 'Cantidad de cargos debe empezar con 0'

rollback tran

end

/\*7 Verificar que al crear un cargo

este cargo inicia con estado inactivo (0)\*/

create trigger estadoInicialCargo

on cargo for insert as

if((select estado

from inserted)!=0)

begin

print 'El estado del cargo debe iniciar como inactivo'

rollback tran

end

/\*8 Actualizar un cargo a estado activo cuando

se inserta a una persona en ese cargo\*/

create trigger actualizarCargo

on personalAdministrativo for insert as

declare @idCarg int

set @idCarg=(select idCargo from inserted)

update cargo set estado=1 where id=@idCarg

/\*9 Actualizar un cargo a estado inactivo cuando

no existan personas en ese cargo\*/

create trigger actualizarCargoB

on personalAdministrativo for delete as

declare @idCarg int

declare cursorCargos cursor for(select idCargo from deleted)

open cursorCargos

fetch from cursorCargos into @idCarg

while(@@FETCH\_STATUS=0)

begin

if((select count(\*)

from personalAdministrativo

where idCargo=@idCarg)=0)

begin

update cargo set estado=0 where id=@idCarg

end

fetch from cursorCargos into @idCarg

end

close cursorCargos

deallocate cursorCargos

/\*10Validar que la persona que realiza

la venta pertenece al departamento de ventas\*/

create trigger validarTrabajadorEnVenta

on notaDeVenta for insert as

if((select departamento.nombre

from personalAdministrativo,cargo,departamento,inserted

where inserted.codigoPersonalAdministrativo=personalAdministrativo.codigo and

personalAdministrativo.idCargo=cargo.id and

cargo.idDepartamento=departamento.id)!='Departamento de ventas')

begin

print 'El trabajador no pertenece al depto. de ventas.'

rollback tran

end

/\*11Crear N cuotas en funcion a su plazo

y a su intervalo de pago\*/

create trigger crearCuotas

on notaDeVenta for insert as

declare @nroNota int

declare @cantidadDeCuotas int

declare @plazoACumplir int

declare @intervalo int

declare @montoPorCuota float

set @nroNota=(select nro from inserted)

set @plazoACumplir=(select plazo from inserted, metodoDePago where inserted.idMetodoPago=id)

set @intervalo=(select idIntervaloDePago from inserted, metodoDePago where inserted.idMetodoPago=id)

set @cantidadDeCuotas=( 12 \* @plazoACumplir ) / @intervalo

set @montoPorCuota=(select monto from inserted)/@cantidadDeCuotas

declare @contador int

declare @cuotasExistentes int

set @cuotasExistentes=(select count(\*)from cuota)

set @contador=1

while(@contador<=@cantidadDeCuotas)

begin

insert into cuota values(@nroNota,@contador+@cuotasExistentes,@montoPorCuota,0)

set @contador=@contador+1

end

/\*12No añadir un grupo que utiliza la misma aula

en el mismo horario de otro grupo\*/

create trigger noAñadirGrupo

on grupo for insert as

if((select count(\*)

from grupo,inserted

where grupo.idHorario=inserted.idHorario and

grupo.nroAula = inserted.nroAula)>0)

begin

print 'No se puede repetir misma aula en mismo horario';

rollback tran;

end;

**MODELO ISHIKAWA**

1. **A**
   1. **Identificar el problema**
      1. **Lista de problemas**

**P1.** El manejo de registros de alumnos es muy deficiente

**P2.** Mal manejo de información de personas interesadas en los cursos.

**P3.** Escases de dinero para la obtención de nuevos objetos.

**P4.** Deudas de salarios a los trabajadores.

**P5.** Personal insuficiente para el cumplimiento de algunas actividades extras.

**P6.** Acumulación y pérdida de formularios de inscripción.

**P7.** Pérdida de tiempo en la búsqueda de formularios a la hora de requerir información sobre un alumno.

**P8.** Los empleados tienen poca información de los cursos ofertados.

**P9.** Mala administración de los cursos.

**P10.** Alto consumo de dinero en material de escritorio.

**P11.** El control de alumnos es muy antiguo y de difícil manejo.

**P12.** Pérdida de documentos por mala organización.

**P13.** Falta de experiencia en cargos de mayor importancia.

**P14.** Personal no capacitado en áreas específicas.

**P15.** Desconocimiento de información de los alumnos.

**P16.** Desconocimiento de información de personas interesadas en algún curso para futuros avisos.

**P17.** La información del instituto no está protegida correctamente.

**P18.** El cumplimiento de los trabajadores es sus respectivas áreas

* + 1. **Depurar problemas**

**P1.** El manejo de registros de alumnos es muy deficiente.

**P2.** Mal manejo de información de personas interesadas en los cursos.

**P3.** Acumulación y pérdida de formularios de inscripción.

**P4**. Pérdida de tiempo en la búsqueda de formularios a la hora de requerir información sobre un alumno.

**P5.** Mala administración de los cursos.

**P6.** Alto consumo de dinero en material de escritorio.

**P7.** El control de alumnos es muy antiguo y de difícil manejo.

**P8.** Perdida de documentos por mala organización.

**P9.** Desconocimiento de información de los alumnos.

**P10.** Desconocimiento de información de personas interesadas en algún curso para futuros avisos.

**P11.** La información del instituto no está protegida correctamente.

* + 1. **Propietarios del problema**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Propietarios>**  **Problemasv** | **ADMINIS**  **TRADORES** | **EJEC.**  **VENTAS** | **GERENTE GENERAL** | **CLIENTES** | **DOCENTES** |
| **P1**El manejo de registros de los alumnos es muy deficiente |  | **X** | **X** |  | **X** |
| **P2**Mal manejo de información de personas interesadas |  | **X** | **X** |  |  |
| **P3**Acumulacion y perdida de formularios de inscripción | **X** | **X** | **X** | **X** |  |
| **P4**Perdida de tiempo en la búsqueda de formulariosa la hora de requerir información sobre el alumno | **X** | **X** | **X** |  | **X** |
| **P5**Mala administración de los cursos |  | **X** | **X** | **X** | **X** |
| **P6**Alto consumo de dinero en material de escritorio | **X** | **X** | **X** |  |  |
| **P7**El control de alumnos es muy antiguo y de difícil manejo |  | **X** | **X** |  | **X** |
| **P8**Perdida de documentos por mala organización | **X** | **X** | **X** |  | **X** |
| **P9**Desconocimien  to de información de los alumnos |  |  | **X** | **X** | **X** |
| **P10**Desconoci  miento de informa  ción de personas interesadas |  | **X** | **X** |  |  |
| **P11**La información del instituto no está protegida | **X** | **X** | **X** |  | **X** |

* + 1. **Análisis del problema**
    2. **Estimación y cuantificación de problemas**

|  |  |
| --- | --- |
| **Propietarios>**  **Problemasv** | **Justificación** |
| **P1**El manejo de registros de los alumnos es muy deficiente | Cuando un alumno nuevo llega al instituto, ya sea para inscribirse o pedir información, éste llena un formulario y dicho formulario es guardado en una pila con los demás formularios, generando un desorden y corriendo un gran riesgo de pérdida. |
| **P2**Mal manejo de información de personas interesadas | Si una persona nueva llega al instituto, y solicita que le den información, deja su número de teléfono, el cual no es aprovechado al máximo por la institución, o lo usan para hacer SPAM, y muchas veces no es del todo agradable. |
| **P3**Acumulacion y perdida de formularios de inscripción | La parte mala de formularios para las inscripciones es la acumulación que se va generando a medida que pasa el tiempo, generando deterioros y a su vez perdidas de documentos. |
| **P4**Perdida de tiempo en la búsqueda de formularios | Cuando se desea informar a antiguos alumnos sobre un nuevo curso relacionado al anterior, la tarea de buscar un número de teléfono puede ser demasiado morosa e incluso llegar a ser inútil por falta de información en un formulario mal rellenado. |
| **P5**Mala administración de los cursos | Si no hay un método de organización, puede llegar a originarse un caos en el tema de cursos, desde una simple asignación de horario, hasta el cambio de aulas de horas de clase de cualquier otro. |
| **P6**Alto consumo de dinero en material de escritorio | Ya que no hay un sistema para el registro de estudiantes, docentes y cursos, se debe realizar todo a mano, incrementando el número de lapiceros a utilizar, así como también lápices, diurex, clips, etc. |
| **P7**El control de alumnos es muy antiguo y de difícil manejo | La forma de registrar a los estudiantes ya debería haber quedado obsoleta, pero por necesidad y por falta de conocimientos en el tema de sistemas para gestionar sus procesos no se ha tocado el tema. |
| **P8**Perdida de documentos por mala organización | Guardar los papeles de cursos, formularios de registros y recibos de ventas, no garantiza un tiempo de vida correcto para su manipulación, e incluso puede llegar al deterioro y perdida. |
| **P9**Desconocimien  to de información de los alumnos | Muchas veces por rapidez al realizar algún proceso, no podemos buscar a información necesaria de ciertas personas, y por ende, se falla en tal proceso. |
| **P10**Desconoci  miento de informa  ción de personas interesadas | No existe un registro de personas que simplemente estén interesadas en algún curso. Se debería pedir un simple número de teléfono con un nombre para almacenarlo un una base de datos. |
| **P11**La información del instituto no está protegida | Los documentos en físico son muy accesibles para cualquier persona extraña al instituto. |

* + 1. **Alternativas de cambio**

**ALT01:** Pedirmás datos a las personas que se las vea interesadas en algún curso del instituto.

**ALT02:** Crear un reglamento sobre los horarios de clases en los cuales se dan los cursos.

**ALT03:** Agilizar los registros de alumnos nuevos mediante un servicio web.

**ALT04:** Guardar los documentos en lugar con restricción de personal.

**ALT05:** Llegar a hacer conocer los cursos a personas más jóvenes interesados en su futuro.

**ALT06:** Realizar informes diarios para control del personal administrativo y académico en el instituto.

**ALT07:** Utilizar un sistema de información para la gestión de cursos, ventas, docentes y alumnos del instituto.

* + 1. **Conclusión de los problemas**

Alternativas seleccionadas:

**ALT01:** Crear un reglamento sobre los horarios de clases en los cuales se dan los cursos. (ALT02)

**ALT02:** Agilizar los registros de alumnos nuevos mediante un servicio web. (ALT03)

**ALT03:** Guardar los documentos en lugar con restricción de personal. (ALT04)

**ALT04:** Realizar informes diarios para control del personal administrativo y académico en el instituto. (ALT06)

**ALT05:** Utilizar un sistema de información para la gestión de cursos, ventas, docentes y alumnos del instituto. (ALT07)

Soluciones:

Al implementar un sistema de información para el instituto:

**P1: P2: P3: P10:**

Estos problemas se solucionarían gracias a que el sistema guardará estos datos y los trabajadores ya no tendrán que hacerlo a mano.

**P4: P5: P7:**

El sistema agilizará las tareas y solucionará estos problemas, reduciendo el tiempo que se perdía y así poderlo usar en otras cosas.

**P6: P8: P9: P11:**

Al mejorar la eficiencia del instituto, este rendirá mucho mejor y asi generará más conformidad en los trabajadores y as u ves un mejor rendimiento en general.

* + 1. **Modelo Ishikawa**

Depto. Gerencial

**P5**

**P6**

Ineficiencia en la gestión de servicios y procesos internos

**P8**

**P10**

**P11**

**P3**

**P1**

**P9**

**P2**

**P4**

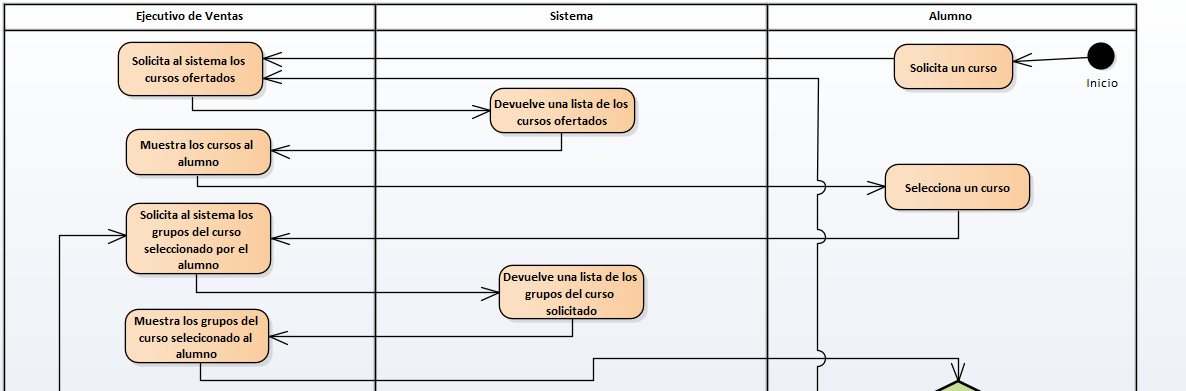
**P7**

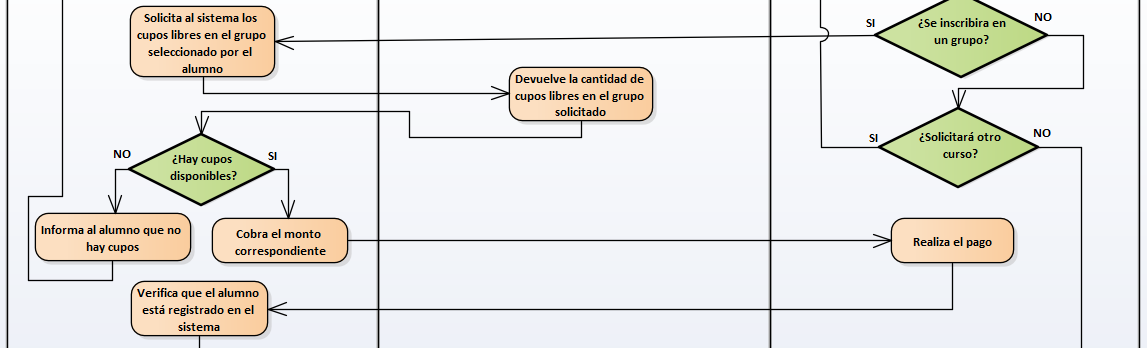
Depto. Académico

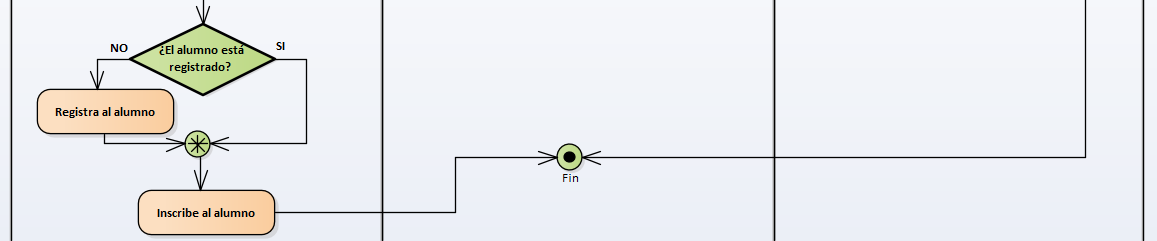
Depto. Marketing

**MODELO DE NEGOCIO**

* 1. **Diagramas de actividad**
* **Venta**

****

****

****

* **Registro de personas**

****

* **Pago de cuota**

****

**CAPTURA DE REQUISITOS**

* 1. **Encontrar actores y casos de uso**
     1. **Lista de actores**
* Alumno: Es la persona que ya está registrada en algún curso, que además puede solicitar otros cursos.
* Docente: Es aquella persona encargada de dar los cursos.
* Personal Administrativo: Es la persona que está encargada de los diferentes departamentos.
* Ejecutivo de Ventas: es la persona que se encarga de la venta de los cursos.
* Administrador: es la persona encargada del registro y supervisión del sistema.
* Sistema:Actúa como un actor pasivo cuando se encuentra en espera a que el administrativo ingrese datos para realizar alguna búsqueda o para almacenarlos en la base de datos.
  + 1. **Lista de casos de uso**

**CU1** Administrar Usuario

**CU2** Administrar Privilegio

**CU3** Gestionar Docente

**CU4** Gestionar Alumno

**CU5** Gestionar Personal Administrativo

**CU6** Gestionar Departamento

**CU7** Gestionar Cargo

**CU8** Gestionar Tipo

**CU9** Gestionar Curso

**CU10** Administrar Grupo

**CU11** Administrar Gestión

**CU12** Administrar Horario

**CU13** Administrar Aula

**CU14** Gestionar Inscripción

**CU15** Gestionar Pago de Cuota

**CU16** Gestionar Cuota

**CU17** Iniciar Sesión

**CU18** Cerrar Sesión

**CU19** Gestionar Administrador

**CU20** Verificar Disponibilidad de Curso

* + 1. **Tabla de actores y casos de uso**

|  |  |
| --- | --- |
| **Caso de uso** | **Actor** |
| CU1 | A5 |
| CU2 | A5 |
| CU3 | A5, A2 |
| CU4 | A5, A1 |
| CU5 | A5, A3 |
| CU6 | A5 |
| CU7 | A5 |
| CU8 | A5 |
| CU9 | A5 |
| CU10 | A5 |
| CU11 | A5 |
| CU12 | A5 |
| CU13 | A5 |
| CU14 | A4, A1 |
| CU15 | A4, A1 |
| CU16 | A4 |
| CU17 | A6 |
| CU18 | A6 |
| CU19 | A6 |
| CU20 | A6 |

* 1. **Priorizar de casos de uso**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caso de uso** | **Estado** | **Prioridad** | **Riesgo** | **Actor** |
| CU1 | Incorporado | Crítico | Crítico | A5 |
| CU2 | Incorporado | Significativo | Crítico | A5 |
| CU3 | Aprobado | Normal | Normal | A5, A2 |
| CU4 | Aprobado | Normal | Normal | A5, A1 |
| CU5 | Aprobado | Normal | Normal | A5, A3 |
| CU6 | Aprobado | Normal | Normal | A5 |
| CU7 | Incorporado | Crítico | Normal | A5 |
| CU8 | Incorporado | Crítico | Normal | A5 |
| CU9 | Aprobado | Normal | Crítico | A5 |
| CU10 | Incorporado | Crítico | Normal | A5 |
| CU11 | Aprobado | Normal | Normal | A5 |
| CU12 | Incorporado | Normal | Normal | A5 |
| CU13 | Incorporado | Normal | Normal | A5 |
| CU14 | Aprobado | Crítico | Crítico | A4, A1 |
| CU15 | Incorporado | Crítico | Crítico | A4, A1 |
| CU16 | Incorporado | Crítico | Crítico | A4 |
| CU17 | Aprobado | Accesoria | Significativo | A6 |
| CU18 | Aprobado | Accesoria | Significativo | A6 |
| CU19 | Aprobado | Normal | Normal | A6 |
| CU20 | Incorporado | Significativo | Crítico | A6 |

* 1. **Detallar un caso de uso**

**CU1 Administrar Usuario**



|  |  |
| --- | --- |
| **Nombre de caso de uso** | Administrar Usuario |
| **Propósito** | Su propósito es que administrador pueda administrar a los usuarios |
| **Resumen** | El sistema permite al administrador del sistema administrar a los usuarios, con cada uno de sus datos personales y poder hacer búsquedas a partir de su cargo |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU3 Gestionar Docente  CU4 Gestionar Alumno  CU5 Gestionar Personal Administrativo |
| **Flujo Principal** | 1. MOSTRAR  1.1 Click en “Ver Usuarios”  1.2 Visualizar datos de los usuarios  1.3 Click en “Next” si es que la lista sobrepasa los 10 usuarios en la lista.  2. BUSCAR  2.1 Click en “Search”  2.2 Escribir el nombre de un usuario para filtrar los resultados  2.3 Visualizar los datos del usuario. |
| **Post Condición** | Ninguno |
| **Excepción** | 2.3 El usuario no existe |

**CU2 Administrar Privilegio**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Privilegio |
| **Propósito** | Su propósito es poder gestionar el privilegio que recibe cada cargo. |
| **Resumen** | El sistema debe ser capaz de asignarle a cada usuario un cargo y ese cargo tener un privilegio. |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU5 Gestionar Personal Administrativo |
| **Flujo Principal** | 1. ADMINISTRAR  1.1. Seleccionar cargo  1.2. Seleccionar privilegios para el cargo  1.3. Dar clic en “Aceptar” |
| **Post Condición** | Ninguna |
| **Excepción** | Ninguna |

**CU3 Gestionar Docente**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Docente |
| **Propósito** | Su propósito es el de permitir gestionar a los distintos docentes. |
| **Resumen** | El sistema permite al administrador registrar nuevos docentes, actualizar sus datos personales y también eliminarlos del sistema. |
| **Actores** | Administrador, Docente |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | Ninguno |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Agregar nuevo Docente”  1.2. Ingresar los datos del nuevo docente:  1.2.1. Carnet de identidad  1.2.2. Nombre  1.2.3. Apellido Paterno  1.2.4. Apellido Materno  1.2.5. Teléfono  1.2.6. Correo  1.2.7. Fecha Nacimiento  1.2.8. Profesión  1.2.9. Nivel de estudio  1.2.10. Lugar de trabajo  1.2.11. Dirección de trabajo  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el docente cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados (los mismos datos que en registrar exceptuando carnet de identidad).  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el docente a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguno |
| **Excepción** | 1.3 Nombres y Apellidos iguales con otro Docente ya registrado.  2.1 No hay Docente registrado.  3.1 No hay Docente registrado. |

**CU4 Gestionar Alumno**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Alumno |
| **Propósito** | Su propósito es el de permitir gestionar a los distintos alumnos. |
| **Resumen** | El sistema permite al administrador registrar nuevos alumnos, actualizar sus datos personales y también eliminarlos del sistema. |
| **Actores** | Administrador, Alumno |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | Ninguno |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Agregar nuevo Alumno”  1.2. Ingresar los datos del nuevo alumno:  1.2.1. Carnet de identidad  1.2.2. Nombre  1.2.3. Apellido Paterno  1.2.4. Apellido Materno  1.2.5. Teléfono  1.2.6. Correo  1.2.7. Fecha Nacimiento  1.2.8. Profesión  1.2.9. Nivel de estudio  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el alumno cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados (los mismos datos que en registrar exceptuando carnet de identidad).  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el alumno a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguno |
| **Excepción** | 1.3 Nombres y Apellidos iguales con otro Alumno ya registrado.  2.1 No hay Alumno registrado.  3.1 No hay Alumno registrado. |

**CU5 Gestionar Personal Administrativo**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Personal Administrativo |
| **Propósito** | Su propósito es el de permitir gestionar a los distintos Personal Administrativos. |
| **Resumen** | El sistema permite al administrador registrar nuevos Personal Administrativos, actualizar sus datos personales y también eliminarlos del sistema. |
| **Actores** | Administrador, Personal Administrativo |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | Ninguno |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Agregar nuevo Personal Administrativo”  1.2. Ingresar los datos del nuevo Personal Administrativo:  1.2.1. Carnet de identidad  1.2.2. Nombre  1.2.3. Apellido Paterno  1.2.4. Apellido Materno  1.2.5. Teléfono  1.2.6. Correo  1.2.7. Fecha Nacimiento  1.2.8. Profesión  1.2.9. Nivel de estudio  1.2.10. Cantidad de hijos  1.2.11. Dirección domicilio  1.2.12. Fecha de ingreso  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el Personal Administrativo cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados (los mismos datos que en registrar exceptuando carnet de identidad).  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el Personal Administrativo a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguno |
| **Excepción** | 1.3 Nombres y Apellidos iguales con otro Personal Administrativo ya registrado.  2.1 No hay Personal Administrativo registrado.  3.1 No hay Personal Administrativo registrado. |

**CU6 Gestionar Departamento**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Departamento |
| **Propósito** | Su propósito es el de permitir gestionar los departamentos del Instituto. |
| **Resumen** | El sistema permite al administrador registrar, modificar y eliminar un departamento si es necesario. |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | Ninguna |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos del nuevo departamento:  1.2.1. Nombre  1.2.2. Cantidad de cargos  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el departamento cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados.  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el departamento a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 1.3. Datos ingresados no válidos  2.1. No hay departamentos registrados  3.1. No hay departamentos registrados |

**CU7 Gestionar Cargo**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Cargo |
| **Propósito** | Su propósito es el de permitir gestionar los cargos del Instituto. |
| **Resumen** | El sistema permite al administrador ir adicionando, modificando o eliminando los diferentes cargos. |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU5 Gestionar Personal Administrativo  CU6 Gestionar Departamento |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos del nuevo cargo  1.2.1. Nombre  1.2.2. Estado  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el cargo cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados.  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el repuesto a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 1.1 No existe departamento para añadir cargo  2.1 No hay cargo  3.1 No hay cargo |

**CU8 Gestionar Tipo**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Tipo |
| **Propósito** | Permite al sistema gestionar los tipos de cursos con sus docentes |
| **Resumen** | Permite el registro, modificación y eliminación de un tipo de un determinado curso con su respectivo docente |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU3 Gestionar Docente  CU9 Gestionar Curso |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos del tipo  1.2.1. Nombre  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el tipo cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el tipo a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 1.3. Ya existe ese tipo  2.1. Tipo no registrado  3.1. Tipo no registrado |

**CU9 Gestionar Curso**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Curso |
| **Propósito** | Su propósito es el de permitir gestionar los distintos cursos |
| **Resumen** | El sistema permite registrar nuevos cursos que se darán, modificarlos y eliminarlos |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | Ninguno |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos del curso  1.2.1. Nombre  1.2.2. Duración  1.2.3. Costo  1.2.4. Cupo Total  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el curso cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el curso a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 1.3. Ya existe ese curso  2.1. Curso no registrado  3.1. Curso no registrado |

**CU10 Administrar Grupo**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Administrar Grupo |
| **Propósito** | El propósito es permitir al sistema administrar los grupos |
| **Resumen** | El sistema permite al administrador tener acceso a los datos de los grupos ya asignados a cada curso |
| **Actores** | Administrar |
| **Actor Iniciador** | Administrar |
| **Pre Condición** | CU3 Gestionar Docente  CU11 Administrar Gestión  CU9 Gestionar Grupo |
| **Flujo Principal** | 1. ADMINISTRAR  1.1. Seleccionar curso  1.2. Seleccionar grupos para el curso  1.3. Dar clic en “Aceptar” |
| **Post Condición** | Ninguna |
| **Excepción** | 1.2. No hay grupo registrado |

**CU11 Administrar Gestión**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Administrar Gestión |
| **Propósito** | Su propósito es el de permitir al administrador administrar las gestiones |
| **Resumen** | El sistema debe ser capaz de permitirle al administrador el acceso a las diferentes gestiones |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | Ninguna |
| **Flujo Principal** | 1. ADMINISTRAR  1.1. Seleccionar grupo  1.2. Seleccionar gestión para el grupo  1.3. Dar clic en “Aceptar” |
| **Post Condición** | Ninguna |
| **Excepción** | Ninguna |

**CU12 Administrar Horario**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Administrar Horario |
| **Propósito** | El propósito es poder administrar los horarios que recibe cada grupo |
| **Resumen** | El sistema le permitirá al administrador asignar los horarios ya establecidos a los diferentes grupos |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU10 Administrar Grupo |
| **Flujo Principal** | 1. ADMINISTRAR  1.1. Seleccionar grupo  1.2. Seleccionar horario para el grupo  1.3. Dar clic en “Aceptar” |
| **Post Condición** | Ninguna |
| **Excepción** | 1.2. No exista horario disponible |

**CU13 Administrar Aula**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Administrar Aula |
| **Propósito** | Su propósito es permitirle al administrador administrar las aulas del establecimiento |
| **Resumen** | El sistema le da al administrador el acceso a los datos de las aulas para poder asignarlas a dichos grupos |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU10 Administrar Grupo |
| **Flujo Principal** | 1. ADMINISTRAR  1.1. Seleccionar grupo  1.2. Seleccionar aulas para el grupo  1.3. Dar clic en “Aceptar” |
| **Post Condición** | Ninguna |
| **Excepción** | 1.2. Todas las aulas estén ocupadas |

**CU14 Gestionar Inscripción**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Inscripción |
| **Propósito** | Su propósito es el de permitir gestionar Inscripciones. |
| **Resumen** | El sistema permite al ejecutivo de ventas generar la inscripción con todos los datos necesarios. |
| **Actores** | Ejecutivo de Ventas, Alumno |
| **Actor Iniciador** | Ejecutivo de Ventas |
| **Pre Condición** | CU10 Gestionar Grupo |
| **Flujo Principal** | 1. SELECCIONAR  1.1. Dar clic en “Generar Nueva Inscripción”  1.2. Seleccionar el curso  1.3. Seleccionar el grupo  1.4. Introducir los datos del alumno  1.3. Dar clic en “Guardar”  2. ELIMINAR  2.1. Seleccionar el servicio a eliminar del sistema  2.2. Dar clic en “Eliminar”  2.3. Confirmar eliminación |
| **Post Condición** | CU20 Verificar Disponibilidad de Curso |
| **Excepción** | 1.4. Los datos son inválidos |

**CU15 Gestionar Pago de Cuota**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Pago de Cuota |
| **Propósito** | Llevar el correspondiente control sobre el pago de las cuotas. |
| **Resumen** | El ejecutivo de ventas podrá llevar el control y asignar los pagos de acuerdo al método de pago. |
| **Actores** | Ejecutivo de Ventas |
| **Actor Iniciador** | Ejecutivo de Ventas |
| **Pre Condición** | CU16 Gestionar Cuota |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos del pago de la cuota  1.2.1. Fecha  1.2.2. Monto  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el pago de cuota cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el pago de cuota a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 1.1.No existe la cuota para realizar los pagos |

**CU16 Gestionar Cuota**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Cuota |
| **Propósito** | Su propósito es el de permitir al administrador gestionar todas las cuotas |
| **Resumen** | El sistema le permite al administrador registrar, modificar y eliminar las cuotas asignadas a cada venta |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU14 Gestionar Inscripción |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos de la cuota  1.2.1. Monto Total  1.2.2. Monto Faltante  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar la cuota cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar la cuota a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 2.1. No existen cuotas registradas  3.1. No existen cuotas registradas |

**CU17 Iniciar Sesión**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Iniciar Sesión |
| **Propósito** | Asegurar que solo personas autorizadas puedan ingresar al sistema. |
| **Resumen** | El sistema puede iniciar sesión a un usuario mediante un nombre de usuario y contraseña introducida por el administrador. |
| **Actores** | Sistema |
| **Actor Iniciador** | Sistema |
| **Pre Condición** | CU2 Gestionar Privilegio |
| **Flujo Principal** | 1. INGRESAR  1.1. Ingresar el nombre y contraseña del usuario  1.2. Dar clic en el botón ingresar |
| **Post Condición** | Ninguna |
| **Excepción** | 1.2. Los datos ingresados son inválidos |

**CU18 Cerrar Sesión**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Cerrar Sesión |
| **Propósito** | Permitir cerrar la sesión de la cuenta de un usuario. |
| **Resumen** | Permitir cerrar la sesión por un usuario en cualquier momento o una vez que haya finalizado alguna acción en el sistema. |
| **Actores** | Sistema |
| **Actor Iniciador** | Sistema |
| **Pre Condición** | CU17 Iniciar Sesión |
| **Flujo Principal** | 1. SALIR  1.1 Dar clic en el botón cerrar sesión  1.2 En la ventana de confirmación seleccionar Aceptar |
| **Post Condición** | Ninguna |
| **Excepción** | Ninguna |

**CU19 Gestionar Administrador**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Gestionar Administrador |
| **Propósito** | Su propósito es el de permitir al sistema gestionar a los diferentes Administradores. |
| **Resumen** | El sistema permite registrar a nuevos Administradores en el caso que exista el Administrador permite modificar. Si ya ni se necesita a un Administrador permite eliminar. |
| **Actores** | Sistema, Administrador |
| **Actor Iniciador** | Sistema |
| **Pre Condición** | Ninguna |
| **Flujo Principal** | 1. REGISTRAR  1.1. Dar clic en “Registrar”  1.2. Ingresar los datos del Administrador  1.2.1. Carnet de identidad  1.2.2. Nombre  1.2.3. Apellido Paterno  1.2.4. Apellido Materno  1.2.5. Teléfono  1.2.6. Correo  1.2.7. Fecha Nacimiento  1.3. Dar clic en “Guardar”  2. MODIFICAR  2.1. Seleccionar el Administrador cuyos datos se van a modificar  2.2. Dar clic en “Modificar”  2.3. Ingresar los datos modificados (mismos datos que en registrar exceptuando el carnet de identidad).  2.4. Dar clic en “Guardar”  3. ELIMINAR  3.1. Seleccionar el Administrador a eliminar del sistema  3.2. Dar clic en “Eliminar”  3.3. Confirmar eliminación |
| **Post Condición** | Ninguna |
| **Excepción** | 1.3 Datos ingresados no válidos.  2.1 No hay Administrador registrado  3.1 No hay Administrador registrado. |

**CU20 Verificar Disponibilidad de Curso**

****

|  |  |
| --- | --- |
| **Nombre de caso de uso** | Verificar Disponibilidad de Curso |
| **Propósito** | Su propósito es permitir que el administrador pueda ver si aún hay cupos disponibles en dicho curso |
| **Resumen** | El sistema permite al administrador del sistema verificar si existe cupo en dicho curso y en cierto grupo e especifico. |
| **Actores** | Administrador |
| **Actor Iniciador** | Administrador |
| **Pre Condición** | CU9 Gestionar Curso  CU10 Administrar Grupo |
| **Flujo Principal** | 1. VERIFICAR  1.1. Introducir datos del curso  1.2. Verifica los cupos del curso  1.3. Verifica los cupos en los grupos del curso  2. SELECCIONAR  2.1 Selecciona el curso y el grupo con cupos disponibles |
| **Post Condición** | Ninguna |
| **Excepción** | 1.1. No existe el curso |

* 1. **Prototipar interfaz usuario**

**ANEXOS**

*Formulario de inscripción para curso de postgrado*



*Formulario de inscripción de curso de capacitación continua*